9 Event Reconstruction

Modern nuclear and particle physics detectors are typically very complex and produce large datasets that must be analyzed **offline** after an experiment is completed. The analysis of the experimental data is typically articulated in two main stages known as **event reconstruction** and **physics analysis**. Basic elements of physics analyses have already been discussed in Chapter 8 and advanced techniques, mostly pertaining to measurements of fluctuations and correlations, will be presented in Chapters 10 and 11. In this chapter, we focus on key aspects of event reconstruction beginning with a brief overview of basic reconstruction tasks in §9.1, and followed in §9.2, with a discussion of track reconstruction techniques, including a detailed example of track reconstruction with a Kalman filter, as well as presentations in §9.3 of various vertex finding and fitting techniques.

9.1 Event Reconstruction Overview

We begin this section with a discussion of the types and structures of data commonly acquired and handled by large experiments in §9.1.1. We next present a general overview of typical data analysis workflows in §9.1.2 and conclude with a brief discussion of basic tasks of detector and signal calibration in §9.1.3.

9.1.1 Types and Structure of Experimental Data

Data produced by large modern experiments typically involve a mix of **slow control data** characterizing the operating conditions (e.g., power supply settings, magnetic field, currents, time, etc.) and performance of the detector, as well as **physics data** produced by the many sensors and detector components of the experiment, as schematically illustrated in Figure 9.1. Physics data include sensor readout in the form of digitized information produced by **analog-to-digital converters (ADCs)** and a variety of **geographical data** indicating the components that produce and report signals. Sensor data may be read out as single values, time sequences, or arrays of various types. They may be **sparsified**, that is, zero suppressed, or read out in their entirety. Speed is usually of the essence and data are assembled by one or several processors (known as **event builders**) into data buffers of various complexities and then dispatched for storage and possibly **online** analysis. Event storage was traditionally accomplished with high-speed magnetic tape drives but is increasingly based on high-performance and high-volume magnetic hard-disk storage systems, as well as optical disks.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

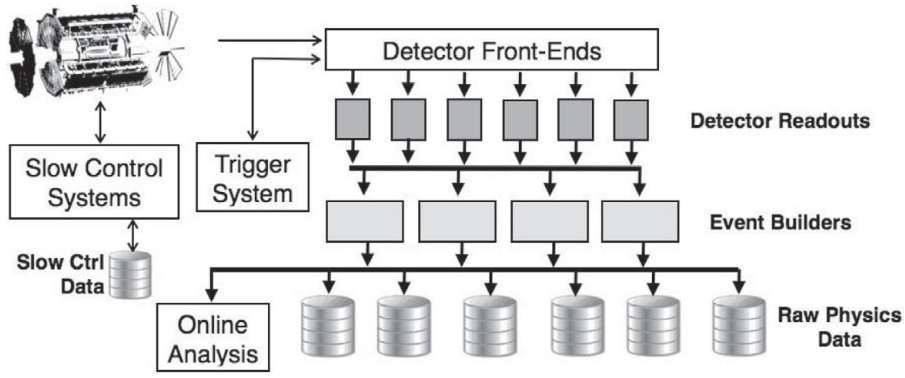


Fig. 9.1 Basic components of modern detector data acquisition systems include detector front-end electronics (analog pipeline, ADC, digital pipelines, sparsification processors), optical readout components, event builder systems, high-speed/capacity storage systems, as well as slow control systems, trigger systems, and an online analysis farm. In this and the following figures, detector diagrams are included for illustrative purposes only. This and Figures 7.2, 7.3, and 7.4 are meant to illustrate the basic concepts rather than actual implementations of the detectors shown.

9.1.2 Data Analysis Workflow

Data produced by large experiments are typically rather complex and usually require multiple stages of analysis, as schematically illustrated in Figure 9.2. There are essentially three basic stages involved in the data analysis of large experiments: (1) event reconstruction,

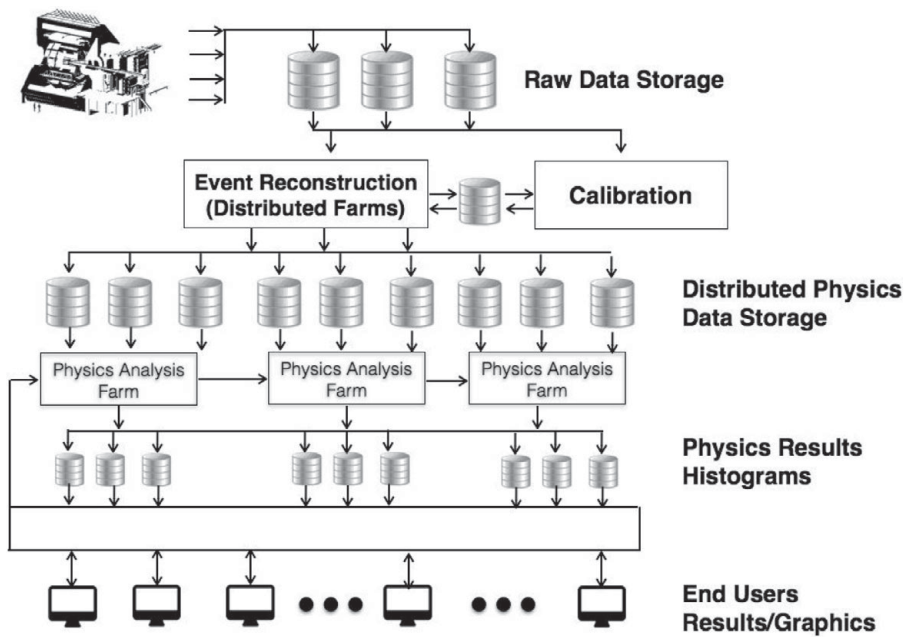


Fig. 9.2 Basic stages of data reconstruction and analysis of large modern experiments.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

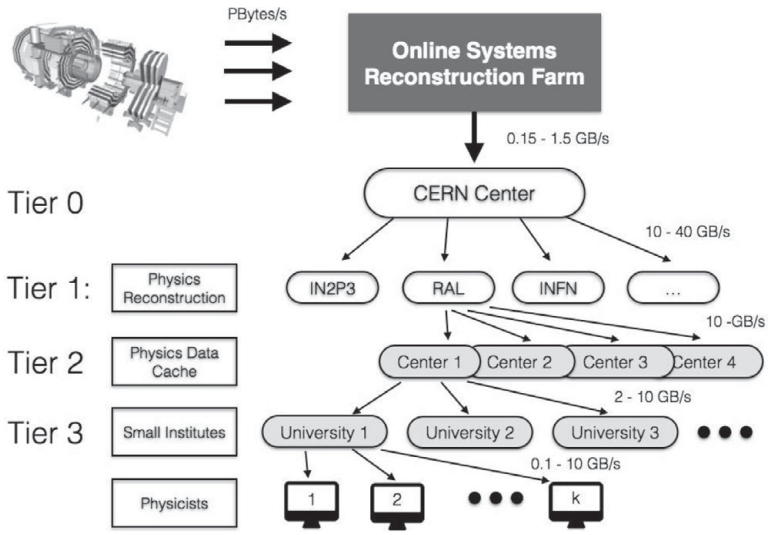


Fig. 9.3 Large-scale modern experiments use multitier grid-based computing architecture to reconstruct and analyze data. Shown here is a schematic layout of grid organization into multiple tiers and stages. Tier 0 and 1 proceed to data calibration and data reconstruction and produce reconstructed events that are then reproduced, archived, and migrated to several Tier 2 centers, where the data are then available for retrieval and analysis by physicists working at their home institution. Different variants of the design shown are used by experiments. Modern large experiments also make increasing use of commercial cloud computing, storage, and data migration.

(2) calibration, and (3) physics analyses. These three stages may be computed at a single facility, or carried out on a multitier computing grid infrastructure, as schematically illustrated in Figure 9.3. Grid computing is now an essential component of large experiments and involves the development and deployment of grid distributed analysis software; Grid management and monitoring; as well as end user tasks that include job submission, monitoring, and result retrieval. A discussion of these concepts and tools lies far beyond the scope of this text. Note, however, that most large experiments have ample documentation describing their specific grid toolset and analysis protocols.

Event reconstruction is usually a time-consuming and resource-intensive task carried out by a dedicated group of scientists (involving physicists and software engineers) on behalf of their collaboration. The main components of the event reconstruction process are schematically illustrated in Figure 9.4, and the object model they are based upon is shown in Figure 9.5. Event reconstruction typically involves the unpacking of the raw data produced by the detector; application of sensor and detector calibrations to obtain physical signals; and various stages of pattern recognition to reconstruct hits, tracks, collision, and decay vertices, as well as other constructs such as V^0 or jets. It may also involve the calculation of various observables deemed representative of the collisions of interest and that are used by most physicists analyzing the experiment's data. The hits, tracks, vertices, and other data entities produced are then assembled into **reconstructed events**, filtered, and stored in easily retrievable data formats (Figure 9.5).

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

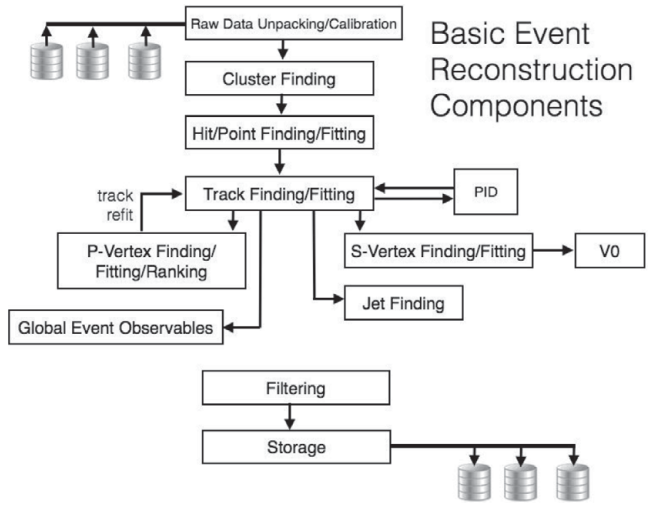


Fig. 9.4 Typical event reconstruction workflow used in modern particle physics experiments.

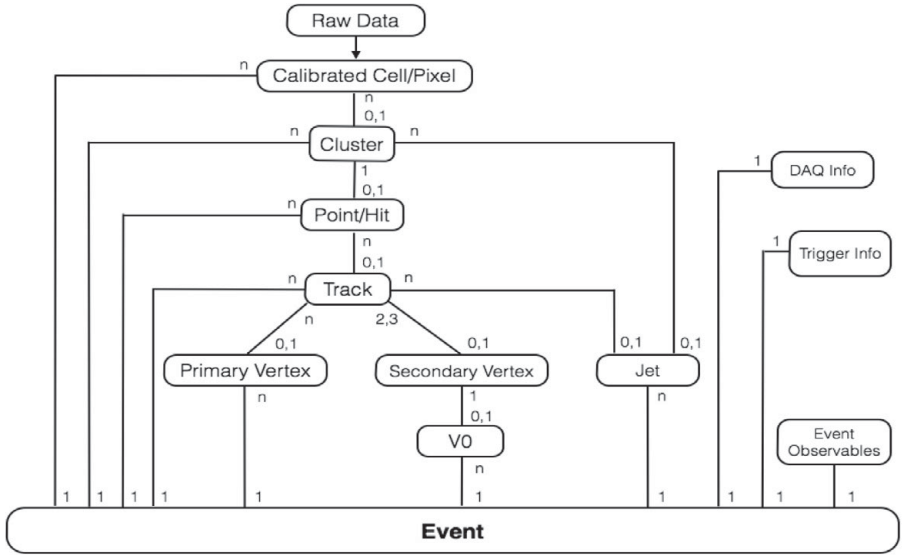


Fig. 9.5 Physics object model and structures commonly used in event reconstruction and physics analyses. Lines and numbers indicate the relation among components of the data model. An *n* indicates several instances of the given type that may be associated with the connected object instance, while 0,1 indicate that at most one instance (one or none) may be involved in the relation. For instance, several tracks, but not necessarily all of them, may be associated with a given vertex. Note that some event reconstruction schemes allow many-to-many, or *n* to *n*, object relations: for example, a point might be allowed to belong to several tracks concurrently, or a track might be associated with several vertices.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

Although the responsibility for event reconstruction is typically reserved for a few experts in each collaboration, it is important to be familiar with the different components and stages of event reconstruction at a conceptual level in order to understand the "features" and limitations of data produced by one's detector.

The tasks of determining the basic kinematic parameters of charged particles at their point of production as well as the location of these production points are commonly called **track reconstruction** and **vertex reconstruction**, respectively. Both these reconstruction tasks may be carried **online**, especially in the high-level trigger, where for the sake of speed, applications frequently use simplified reconstruction methods. However, our discussion will concentrate on methods usually employed in offline analyses, where precision and quality are the major goals, while speed is rarely a decisive factor. Track and vertex reconstruction components are arguably the most important, and we consequently dedicate two sections of this chapter to these tasks. It should be understood, at the outset, that our discussion of these topics is meant to be illustrative of the methods and computations involved only, and as such cannot cover the vast array of methods and techniques that have been developed in the subfield of nuclear and particle physics. Readers particularly interested in these reconstructions should consult the recent literature on these topics, and most particularly recent advances by the CMS Collaboration on the particle flow algorithm [34].

The calibration stage typically involves a plurality of tasks that operate either on raw unpacked data or physical variables derived from them. Virtually all signals produced by a detector must be calibrated in some fashion. Calibration tasks indeed range from gain calibration of charge, light, or energy sensors, calibration of timing signals, as well as geometry and detector alignment. Some of the basic calibration tasks involved in modern experiments are presented in §9.1.3. Here again, we emphasize that it is impossible to cover all types of calibration procedures used in experimental measurements. We thus limit our discussion in §9.1.3 to a few illustrative examples only.

The physics analysis stages involves a wide variety of computing tasks that are often segregated in different groups based either on the data employed or the analysis techniques used to extract physics results. Groups often produce sub-datasets consisting of a fraction of events or distilled events that store only a subset of observables. This enables faster and more efficient analysis of the data of interest. This said, at this stage speed and computing efficiency, although important, are usually not a decisive factor, and analyses may be repeated numerous times with various tweaks of control parameter values to extract physical observables of interest and study systematic errors (discussed in detail in §12.5).

9.1.3 Signal Calibration

Particle detectors measure a wide variety of physical observables, ranging from energy deposition in calorimeters, gas cells, or pads, or solid-state devices, currents, and voltages, timing signals including those of time-of-flight detectors and clocks, various types of registers and hit arrays, and so on. All detector analog signals are digitized and recorded for future analyses. A vast number of techniques and technologies have been developed for signal amplification, filtering, analog storage (pipeline), digitization, sparsification, event building and packing, and data storage. A discussion of such topics is beyond the scope

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

of this text but may be found in various other publications. We here restrict our discussion to the main types of digital signal processing required to extract physically meaningful information out of digitized data.

Analog-to-Digital Conversion

An analog-to-digital converter, or ADC, is a device that converts a physical signal, such as a voltage or an integrated charge, to a digital number that represents the quantity's size or amplitude. Conversion of signals into digital numbers involves quantization and digital representation of the signals with finitely many bits. The number of bits used in analog-to-digital conversion is determined by the needs and specificities of the physical signals being handled. For instance, a single bit, 0 or 1, is sufficient to indicate that a signal has crossed a minimal threshold, but several bits are needed to provide meaningful resolution in the digitization of energy deposition or timing signals. The conversion from a continuous-time or continuous-amplitude analog signal thus results in a sequence of discrete-time or discrete-amplitude digital signals that feature a finite granularity. Quantization thus involves loss of information and the introduction of quantization errors. Such errors may, however, be rendered negligible by judicious choices of dynamic range and number of bits used in the conversion.

ADCs can fulfill a wide variety of purposes and thus have a wide range of design and performance characteristics:

1. Bandwidth: The bandwidth of an ADC corresponds to the range of signal frequencies it can measure.
2. Number of bits: The number of bits used in the analog-to-digital conversion.
3. Signal-to-noise ratio: How accurately an ADC can measure a signal relative to the noise and information loss it introduces.
4. Sampling rate: The speed at which an ADC can carry out conversions.
5. Aliasing: How an ADC handles the errors introduced by digitization.
6. Dynamic range: Ratio between the largest and smallest possible values an ADC can convert.
7. Linearity: The degree to which there is a linear relation between the analog and digitized signals.
8. Accuracy: The degree to which quantization levels match the true analog signal.
9. Effective number of bits (ENOB).

The bandwidth of an ADC is determined primarily by its sampling rate, and to a lesser extent by how it handles errors such as aliasing, whereas the dynamic range is largely determined by its resolution, that is, the number of "levels" it can convert a signal to, its linearity, and its accuracy. The dynamic range is often summarized in terms of an effective number of bits (ENOB), the number of bits produced that are on average not noise. An ideal ADC has an ENOB equal to its resolution.

The design of ADCs is usually constrained by the desire to match the bandwidth of the signal and the required signal-to-noise ratio as well as engineering (e.g., size, power

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

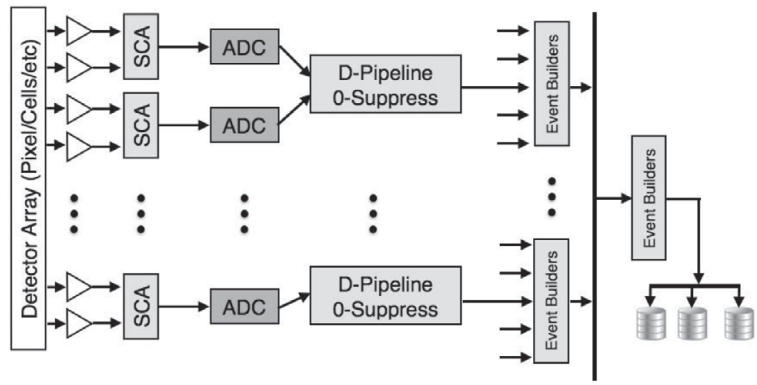


Fig. 9.6 Schematic illustration of the stages involved in detector signal read out and digitization.

required, etc.) and cost considerations. An ADC operating at a sampling rate greater than twice the bandwidth of the signal can in principle achieve perfect signal reconstruction, but the presence of quantization error limits the dynamic range of even an ideal ADC. This said, if the dynamic range of an ADC exceeds that of its input signal, the effective quantization error may be negligible, thereby resulting in an essentially perfect digital representation of the input signal.

The input of an ADC may consist of a single signal to be digitized once a trigger is received, or involve a continuous signal stream to be sampled and processed uninterruptedly. Signals may also be received and processed from multiple interleaved streams or from switched capacitor arrays used to hold analog signals from one or several signal streams until a trigger is received (see Figure 9.6). Consequently, the digitized output may consist of a single number or a stream of numbers to be stored directly into some computer memory or held momentarily onto a digital pipeline device. Digital pipeline devices are used, in particular, toward the sparsification of data, that is, the elimination of null or vanishing signals as well as triggering purposes, thereby considerably reducing the volume of data produced collision by collision.

ADCs can be used to convert voltage amplitude, integrated charge (charge-to-digital conversion, often called QDC), as well as timing signals consisting of start and stop logical pulses (time-to-digital conversion, or TDC). Many ADC designs and implementations have been developed over the years that match the wide range of scientific, engineering, and commercial applications in use today [90, 134, 186].

Basic Digital Signal Analysis

Pedestal Determination and Removal

No matter what technology is used to build them, ADCs are usually designed to produce a non-null number, called a **pedestal**, even in the presence of a vanishing signal (see Figure 9.7). This enables the identification of slightly negative signals as well as determination of the noise characteristics of the detector signals being converted. Effectively, the production of a pedestal is achieved by leaking a finite amount of charge or current at the input. In the

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

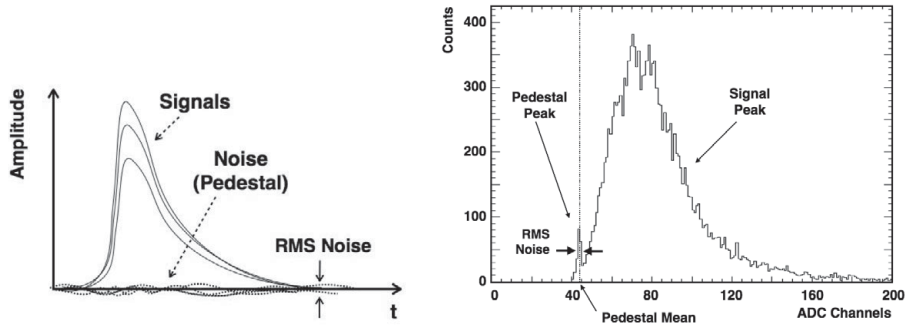


Fig. 9.7 Illustration of the notion of ADC pedestal: A small but finite current is fed into an ADC's input in order to produce a digital output with constant nonzero offset known as a pedestal. The RMS width of the pedestal peak is determined primarily by the noise characteristics of the readout chain.

presence of a truly vanishing and noise-free signal, this should ideally produce a pedestal value consisting of a single number. In practice, ADC pedestals may have a finite width, which is usually characterized in terms of a root mean square number of bins (levels), as schematically illustrated in Figure 9.7. Obviously, the incoming signal has a small quiescent current and noise of its own, which usually broadens the width of the pedestal and may also change its position.

The determination of the centroid and width of ADC pedestals is thus the most basic and elementary operation to be conducted on measured signals. The evaluation of pedestals requires data be accumulated and recorded in the absence of beams (collisions) and with veto against spurious events such as cosmic ray detection. Given a sample consisting of n signals s_i produced by a single ADC, it is usually sufficient to obtain the expectation value, s_0, and standard deviation, s_{std} of these signals to fully characterize a given ADC channel:

$$s_0 = \frac{1}{n}\sum_{i=1}^{n} s_i, \tag{9.1}$$

$$s_{std} = \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(s_i - s_o)^2}. \tag{9.2}$$

The average pedestal value s_o is used in the determination of calibrated signals discussed in the text that follows, whereas s_{std} serves as a basic characterization of the resolution that can be achieved with the given signal.

The performance of ADCs is known to "drift" over time. While their gains are typically quite stable, pedestals are usually observed to shift by few ADC channels over periods ranging from several hours to few days. It is thus necessary to recalibrate ADC pedestals on a regular basis, over a period commensurate with the rate at which they are observed to drift in practice. It is also important to note that AC couplings may lead to 60 Hz (or other frequencies such as the frequency of power supplies) leakage resulting in correspondingly low frequency variations of the pedestal values. Such variations can in principle be eliminated or suppressed by proper grounding of the electronics. However, one can also use

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

an empty ADC slot to monitor the AC couplings in real time and effect a correction based on the phase of the signal measured by the empty ADC slot. More sophisticated techniques are also possible when couplings involve a large number of ADC signal streams (see, e.g., [152]).

Large modern experiments typically make use of dedicated electronic components (e.g., ASICs) mounted directly onboard with the amplification + ADC electronics to accomplish semi-automated pedestal determination and storage. They often also employ a variety of sparsification techniques to eliminate null data. Physicists carrying out analyses of data produced by large modern experiments thus typically do not handle the tasks of pedestal determination and monitoring.

Gain Calibration

The voltage signals produced by detectors are typically very weak and thus require amplification, which may take place within the detection device itself (e.g., gas amplification) or "external" components involving typically two steps (preamplifier and an amplifier). One way or another, the signals s_i produced by ADCs typically have a somewhat arbitrary amplitude which must be calibrated before they can be of any use in physics analyses. Procedures for gain calibration are as varied as the devices used in modern experiments. It is thus not possible in the context of this textbook to offer a full and comprehensive discussion of all techniques used toward gain calibration. However, it can be pointed out that calibration can in general be achieved by measuring a precalibrated signal, that is, a signal of known strength or value. For calorimeters or other types of energy-sensing devices, this can be done by exposing detector modules to beam particles of specific and known energy, cosmic rays, or through the use of radioactive sources of precisely known activity. Various calibration techniques are also based on prior knowledge of the mass of short-lived particles. It is indeed possible to use the known mass of narrow resonances (e.g., ϕ-meson) to obtain energy or momentum calibrations. Techniques to accomplish such calibrations are abundantly documented in the scientific literature (see, e.g., [24, 76, 79, 190]).

Calibrations can be determined either in absolute terms (and units) or relative to convenient references easily obtained or replicated in an experimental environment using, for instance, calibrate laser signals or physics data.

Given a raw ADC signal, s_i, the calibrated signal, E_i (e.g., an energy signal) produced by a specific data channel (i.e., corresponding to a specific detector module or component) is obtained according to the expression

$$E_i = g_i \times (s_i - s_o) \qquad \text{Linear response} \tag{9.3}$$

where s_o is the average pedestal value of the given module i.

This simple linear relation assumes the detection, amplification, and digitization processes are all linear. If either of these components fails to be perfectly linear, a detailed study of the response of the "faulty" component(s) is required, and one must apply a gain correction that is channel- and signal-strength specific:

$$E_i = g_i(s_i) \times (s_i - s_0) \qquad \text{Nonlinear response.} \tag{9.4}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

It should be noted that large (voltage) signals fed into an ADC might exceed its dynamic range. This may be indicated either by a special bit being set to one (1) or as a specific saturation value. The preceding calibration equation fails for inputs exceeding the saturation threshold of an ADC, and therefore, one must be cognizant of the mechanism used by a specific type of ADC to indicate its signal saturation and handle such signals accordingly. Additionally, note that some ADCs are designed to have dual gain functions: a voltage comparator is first used to evaluate the amplitude of the incoming signal and decide which of two gain settings should be used in the digitization of the signal. This produces ADCs with a large effective number of digitization bits, but usually at much lower cost. Obviously, in the calibration and analysis, one must check which of the low- or high-gain modes was used to digitize a specific signal so the appropriate gain g_i may be used to produce the calibrated signal E_i.

Timing Signal Calibration

A **time-to-digital converter (TDC)**, is an ADC designed to convert a time interval into a digital binary output. The time interval to be converted is usually specified by two events (or signals) in the form of logical pulses. TDC devices, much like regular ADCs, usually incorporate several channels and typically operate either on the basis of a common start or common stop (Figure 9.8). In common start mode, a signal produced by a dedicated trigger is used to start the TDC clock. Each channel input into a TDC module then supplies a stop time corresponding to the time elapsed between the trigger and the detection of a signal (above a preset threshold) in a certain detector module. Alternatively, in common stop mode, detector modules each produce their own start signal and the time-to-digital conversion stops when a common stop signal is received by the TDC. In common stop mode, an arming signal may have to be supplied to ready the TPC for conversion before individual starts are received. Whether in common start or common stop mode, the lack of an input signal (stop or start) is indicated by a time-out or saturation bit.

Basic TDC implementations use a high-frequency counter (oscillator) to increment clock cycles. Counting is initiated and stopped when receiving start and stop signals, respectively. An analog method using a voltage ramp and comparator is used for very short time intervals for which a high-frequency counter cannot be used. Either way, one must realize that the measured time interval may not exactly correspond to the time interval between events of interest (e.g., a particle traversing start and stop counters) because the cabling used to carry both the start and stop signals to the TDC has, in general, arbitrary lengths, which add delays to the instantiation of both the start and stop signals. These arbitrary delays must consequently be calibrated out.

Calibrations in common start or common stop modes are essentially carried out in the same way. In essence, one needs to identify either an external time reference that provides an absolute time calibration (e.g., using calibrated laser pulses) or a data-based indicator (e.g., particles traveling nearly at the speed of light on a known flight path) which enables the determination of what is effectively a time pedestal, t_0, but is commonly called "time zero offset" or simply "time zero." The start-to-stop time, which might correspond to a particle's time-of-flight (TOF) between two specific detector components, may then be

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

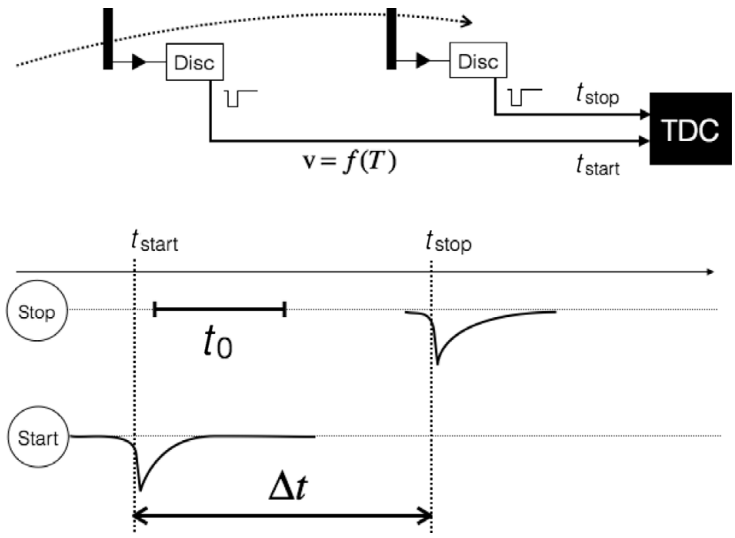


Fig. 9.8 Illustration of timing measurements and the notion of time zero offset.

written

$$t = t_{\text{stop}} - t_{\text{start}} + t_0. \tag{9.5}$$

Few remarks are in order. First, note that the precision (resolution) δt achieved on t depends on the resolution of both the start and top signals:

$$\delta t = \sqrt{\delta t^2_{\text{start}} + \delta t^2_{\text{stop}}}. \tag{9.6}$$

However, the accuracy of the result, that is, whether there is a bias, also depends on the value of t_0. Indeed, if t_0 is not properly calibrated, t will, on average, be either over- or underestimated. All quantities derived from t, such as particle speeds and masses, will thus also be biased.

TOF resolutions required and achieved in modern high-energy experiments range between 50 and 100 ps. Such performances are typically achieved with TOF detectors based on MRPCs or ultrafast scintillator materials (see §8.4.2). The resolution achieved depends on the design and quality of the MRPC or scintillation materials: the stability of amplification, discriminators, and cabling components used in the generation and propagation of timing signals; as well as the resolution and stability of the TDC used for time-to-digital conversion (see Figure 9.8). Much like ADC pedestals, the time calibration t_0 is often observed to slowly drift over time owing to changes in sensor or amplifier gain or due to slight temperature variations. Timing delays produced by cabling or electronic components may indeed change over periods of several hours as a result of temperature variations, for instance. They thus produce slowly varying timing signal drifts that must be compensated for by recalibrating the value of the t_0 of the TOF components on a time scale commensurate with the time scales over which such variations take place.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

Position Calibration

The determination of hit positions, that is, the locations where particles produce energy deposition in detector components used in the reconstruction of trajectories, often rely on weighted sums such as

$$x_{\text{hit}} = \frac{\sum_i E_i^{\alpha} x_i}{\sum_i E_i^{\alpha}}, \tag{9.7}$$

where the sums are taken over the same range of module or components, with x_i and E_i representing the nominal positions and energy read out by these components, respectively. The position x_{hit} is commonly referred to as **centroid** of the energy deposition. An exponent value $\alpha = 1$ is typically appropriate for segmented readout detectors with a small dynamic range (e.g., position sensitive detectors such as silicon strip detectors) but nonlinear weights ($\alpha \neq 1$) are typically more appropriate for segmented calorimeter read-outs. The precision and accuracy achieved in the reconstruction of hit positions depend on proper measurements of both the energies E_i and the position x_i of the components as well as optimum selection of the exponent α (either derived from detailed simulations or data driven analysis of the detector response). These measurements consequently depend on the accurate calibration of signal pedestals and gains, discussed earlier in this section, proper determination of the position of the components (geometrical calibration), as well as precise modeling (studies) of the energy response of the detector modules.

While the design and construction of detector components should in principle dictate their positions relative to one another and relative to the beam axis, target, or interaction region, deviations from the nominal design may occur in practice because of a variety of circumstances including, improper detector placements, shifts in positions due to temperature variations, or intense magnetic fields. It is thus important to properly "calibrate" the position of the detector components before attempting precise data analyses.

Precise determination of detector component positions involves a variety of techniques, including in situ component surveys, laser-based measurements, as well as a host of **detector alignment** techniques based on physics data. Detector alignment techniques based on data rely on the fact that charged particles, in particular, travel on average on well-defined trajectories that can be modeled with precision. Analysis of systematic deviations of layer positions relative to measured particle trajectories thus enables fine adjustments of the component's positions and proper detector alignment. A discussion of techniques used in detector alignment is beyond the scope of this textbook but may be found in various publications [47, 111, 179].

9.2 Track Reconstruction Techniques

Charged particle tracks produced by particle detectors are the basis of much of the analyses carried out in modern nuclear and particle physics. Given that trajectories of charged particles may be significantly altered by the media they traverse largely as the result of

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

multiple coulomb scattering, it is important to use detector technologies that are as thin as possible. Quite a few such technologies have been developed over the years, and very high momentum resolution detectors are now routinely used in the study of particles produced by elementary nuclear collisions. However, there remains the task of identifying these trajectories (hereafter called **tracks**) based on recorded data in order to obtain the kinematic parameters of the particles that produced them. This is the domain of the subfield known as **track reconstruction**, traditionally divided into two stages called **track finding** and **track-fitting**. Track finding involves processes of pattern recognition and classifications, while track-fitting aims at extracting the kinematic parameters of tracks at their point of origin with as much precision as permitted by recorded data and the hardware technologies on which they are based.

A large number of mathematical techniques and algorithms have been developed for both the finding and fitting stages of track reconstruction. There are even techniques that gracefully integrate the two tasks in a single process. We discuss track finding and pattern recognition commonly used with large detectors in §9.2.1 and elaborate on high-efficiency and precision track-fitting techniques in §9.2.2. The underlying theory of how charged particles interact with matter and the basic detection principles used in modern experiments are described in several publications [20, 50, 135] and are not covered here.

9.2.1 Track-Finding Techniques

Track finding is a pattern recognition process that consists of associating recorded hits produced by particles traversing a detection system into groups or classes believed to originate from a common particle. Associated hits form a **track candidate**, which must then be fitted and tested in order to determine whether it constitutes a genuine track or a fake track resulting from incorrect associations of actual or fake hits. Track finders are usually designed to be rather conservative and thus tend to produce and keep a large number of track candidates: eliminating track candidates after the fitting stage is relatively straightforward, while a track candidate discarded at an early stage is not recoverable at the fitting stage. This said, many finders use an iterative procedure involving several track-finding passes either with a single or several distinct algorithms.

Unambiguous track reconstruction and identification is readily achievable when the distance between tracking planes is considerably smaller than the typical distance between hits on a given detection plane. Tracking in a high hit density environment where this condition is not verified may be nonetheless possible if the origin and overall topology of tracks is known, as schematically shown in Figure 9.9.

Track-finding techniques are broadly divided into two classes commonly referred to as **global** and **local** methods. Global methods process all hits "simultaneously," whereas local methods proceed through lists of hits sequentially and build tracks iteratively. Conformal mapping, Hough transforms, and Legendre transforms, presented in the following paragraphs, are examples of global methods, whereas track road and track-following methods, discussed later in this section, are local methods. Additionally, some finding techniques require a track model and prior knowledge of the propagation of particles within the detection system. Others use combinatorial or hit proximity criteria. The use of simplified track

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

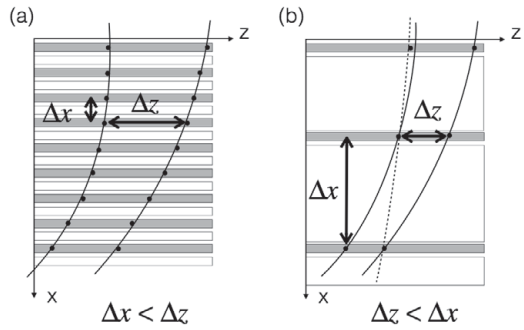


Fig. 9.9 Problematic of pattern recognition and track reconstruction. (a) Track identification is readily achievable when the distance between tracking planes is considerably smaller than the typical distance between hits on a given detection plane. (b) Ambiguities arise in the reverse situation but may be in some cases avoided if the origin or other external constraints exist on the shape and momentum of the track.

models and fast-tracking methods are important in online applications, particularly those where track finding is an integral part of the event-triggering process. We will not consider such applications and refer the reader to specialized publications on this topic. We here focus on techniques used in offline analysis.

A track may be fitted after all its hits have been identified (global track fit) or iteratively as successive hits are added to the track. The Kalman filter and adaptive techniques introduced in §9.2.3 constitute the basis for the best and most robust track reconstruction techniques available to date. Given their mathematical complexity and the fact they are computationally rather intensive, simpler techniques are often used in practice to effect a rapid reconstruction of events. Global track reconstruction techniques such as conformal mapping, Hough transforms, and track templates may be considerably faster and easier to implement. However, they typically require vast amounts of computer memory and fare poorly for secondary tracks.

It is important to note that proper reconstruction of tracks and their kinematic parameters require the tracking device be accurately calibrated. This means that their physical positions and the procedures used to extract hit positions must be accurately determined. An important component of the calibration of hit-sensing devices is a procedure known as **detector alignment**. A discussion of detector alignment techniques is beyond the scope of this textbook but may be found in the relatively recent review of Strandlie and Frühwirth [178].

Adaptive fit methods incorporate competition between several (track) hypotheses, such that the outcome of the competition depends on the current observations. They also feature the possibility of making soft decisions and of combining hypotheses according to their posterior weights. Some adaptive methods have a strong Bayesian character, reflected for example in the presence of posterior probabilities or weights attached to the different hypotheses. In general, weight values change after the inclusion of additional information from the data. Methods can also exhibit different degrees of adaptivity, depending on the level of prior assumptions inherent to the actual approach. Good adaptive methods achieve maximum flexibility and robustness with as few assumptions about the data as possible.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

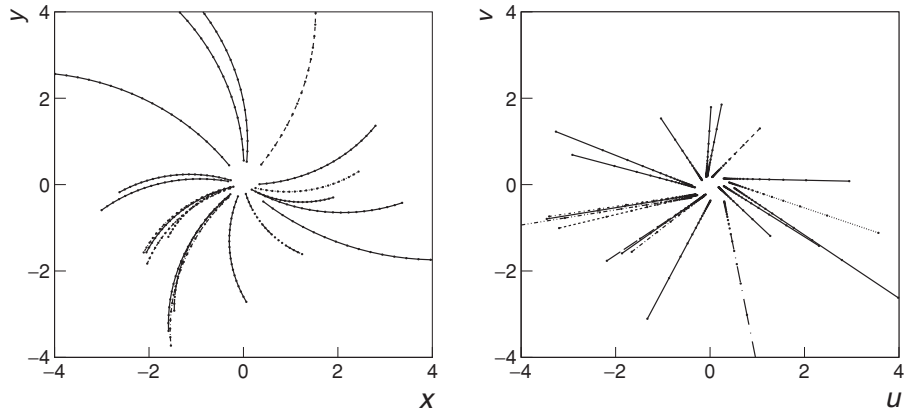


Fig. 9.10 Track finding by conformal mapping $(x, y) \rightarrow (u, v)$.

Conformal Mapping

The track-finding and fitting conformal mapping method [102] is based on the notion that circles going through the origin of a two-dimensional x–y coordinate system map onto straight lines in a u–v coordinate system defined by the conformal transformation:

$$u = \frac{x}{x^2 + y^2} \tag{9.8}$$

$$v = \frac{y}{x^2 + y^2} \tag{9.9}$$

where the circles are defined by

$$(x - a)^2 + (y - b)^2 = r^2 = a^2 + b^2 \tag{9.10}$$

One can verify (see Problem 9.2) that straight lines in the u–v planes may be written as

$$v = \frac{1}{2b} - \frac{a}{b}u \tag{9.11}$$

As illustrated in Figure 9.10, for large-momentum tracks (i.e., large values of the radius r) the conformal mapping yields straight lines passing relatively near the origin of the u–v coordinate system, which can be easily identified. Fits to straight lines in u–v space are fast and yield parameters a and b which, when combined according to Eq. (9.10), provide the radius of a trajectory and its center.

Hough and Legendre Transforms

Conformal mapping, presented in the previous section, is useful when the experimental interest is focused on large momentum particles but becomes less practical for low-momenta tracks which yield straight lines not passing through, or near, the origin of the u–v plane. The Hough transform [109] provides a useful remedy to this problem.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

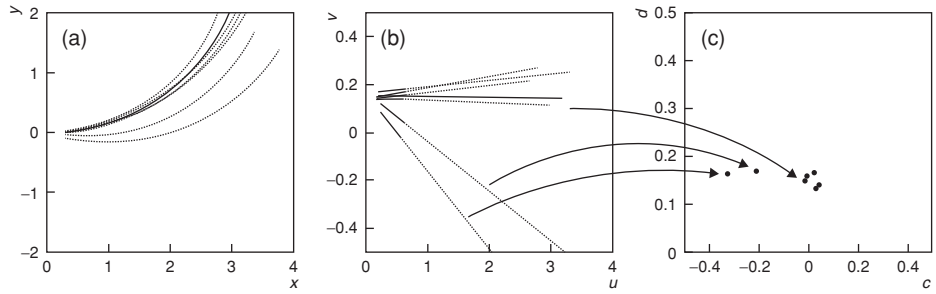


Fig. 9.11 Example of the Hough transform used in the reconstruction of tracks. (a) Hits from charged particles in the bending plane (x–y). (b) Conformal mapping of the hits onto u–v space. (c) Hough transform onto c–d space. Hits belonging to given tracks yield peaks, here represented as solid dots.

The Hough transform is based on a simple mapping of the equation of a straight line in the u–v plane, written $v = cu + d$ to another straight line in the c–d plane, $d = -uc + v$. This means that points along a line in the c–d plane correspond to all possible lines passing through the point (u, v) in the u–v plane. Points located on a straight line in the u–v plane thus tend to create peaks in the c–d plane, which cross at the point that specify the actual parameters of the line in the u–v plane. One can then discretize the c–d plane and increment a histogram for each hit value (x, y). Peaks in this space therefore identify curves in the x–y space. As illustrated in Figure 9.11, one gets a two-dimensional histogram that can be quickly scanned for peaks, which correspond to circular tracks in the x–y plane.

The Hough transform can be adapted to track finding with drift tubes using Legendre transforms [18]. It has also been extended to a variety of pattern recognition problems.

Road Finders

Road finders, also called track template finders, are an example of a global finder technique based on specific track models. The idea is to map each measured point or hit onto all of the track roads (or track templates) to which they could belong. Once all hits of an event have been processed, one then counts how many hits each track template gets. Track candidates are then those templates that have a required minimum number of entries, as illustrated in Figure 9.12. The track candidates must next be fitted to obtain precise kinematic parameters and test their quality.

The road finder technique is quite useful and efficient for finding primary tracks with a specific vertex point. However, it becomes computationally onerous for searching secondary tracks because the number of roads to be considered is likely to be extremely large. In very large track multiplicity environments with a high yield of secondaries, track-following techniques are then more likely to be effective.

Track-Following Technique

The **track-following** technique is a rather generic track-finding method that can be applied with or without track models. Its use in the absence of a track model is possible in a low

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

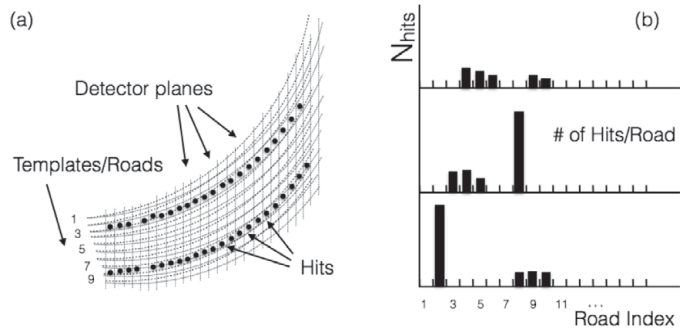


Fig. 9.12 Schematic illustration of the road finder technique. All hits in all detector planes are scanned. For each hit, one increments all bins corresponding to the road they cross. Tracks are those roads with the largest number of hits.

hit density environment and requires the distance between tracking planes be much smaller that the typical distance between hits on a given plane. However, the use of track models enables track reconstruction in much higher density environments.

Track following starts with a **track seed** and proceeds iteratively to the extension of the track, as illustrated in Figure 9.13. Seeds can be formed in the inner region of the tracking detector (e.g., close to the interaction region) if high precision hit measurements are available and the hit density is sufficiently modest. The finder is then described as an inside-out finder. Alternatively, the seed may be first constructed in the outer region of the detector, where hit and track densities are usually much lower. Track extension proceeds in three steps: extrapolation to the next layer, search for matching hits, and choice of best

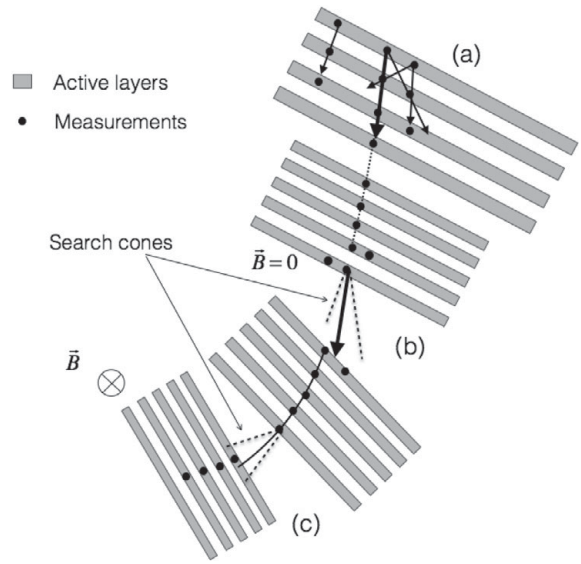


Fig. 9.13 Schematic illustration of the track-following technique. (a) Formation of track seeds in the outermost layers of the detection system. (b) Linear extrapolation in field free regions. (c) Circular extrapolations in finite field regions.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

matching hit if multiple matching hits are found. If no hit is found on a particular layer, the extrapolation is carried to the next layer. The extension procedure is iterated until the number of layers with a missing hit exceeds a preset threshold or until the boundary of the detection system is reached.

9.2.2 Track-Fitting Techniques

The track-fitting stage is subject to challenging constraints. It should be as computationally fast as possible, particularly in the study of relativistic heavy-ion collisions that produce thousands of tracks,[1] but it should not sacrifice the quality of reconstruction. It should be robust against mistakes (e.g., inappropriate hit association and missed hits) made during the track-finding procedure; it should be numerically stable and as accurate as possible based on the precision of the hits supplied by the tracking detectors. It should also be possible to explicitly test whether a track candidate is a valid track. The detection and addition of hits to a track are stochastic processes. Track finding and fitting are thus bound to have limited efficiency and accuracy and may be hampered by the inclusion of hits that do not actually belong to the track.

The testing of tracks can be accomplished, for instance, on the basis of the χ^2 obtained during the fit. The χ^2 is determined by the sum of the squared standardized differences between the measured positions of the track candidate and the estimated positions of the track at the points of intersection of the detector devices. If the χ^2 is very large (given the number of degrees of freedom), the set of associated hits is not statistically compatible with the hypothesis of having been produced by a single particle. The incompatibility may arise from a number of circumstances: for example, extraneous hits might have been wrongly added to the track candidate or a totally illegitimate candidate was built from a random set of hits produced by several distinct particles. Sometimes, the incompatibility results from detector noise or background (e.g., ghost tracks). An outlier hit removal stage and track refit are typically included in track-fitting algorithms. Other track selection criteria are often applied (see §8.3.1). The covariance matrix obtained from the fit additionally provides an estimate of the uncertainty of the track kinematic parameters.

A few elementary considerations are in order, including the choice of a track parameterization, track propagation model, error propagation, effect of detector materials, and track seed finding. These are discussed in following paragraphs of this section.

Track Parameterization

While the goal of track reconstruction consists in the determination of kinematic parameters $\vec{p} = (\vec{p}_T, \phi, p_z)$ of a particle at its point of production (i.e., a primary or secondary vertex), the reconstruction process and association of a track to its proper production vertex (see §9.3.1 for a discussion of vertex finding and track association to vertices) require accurate determination of its actual trajectory through the detection apparatus, which involves finding all hits belonging to a track.

[1] Or in the context of LHC run 3 expected to deliver unprecedented data volumes.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

The trajectory of a charged particle in a static magnetic field $\vec{B}(\vec{r})$ is determined by the Lorentz force:

$$\frac{d\vec{p}}{dt} = \frac{d(m\gamma\vec{v})}{dt} = kq\vec{v}(t) \times \vec{B}(\vec{r}), \tag{9.12}$$

where $\vec{v}$ is the velocity of the particle, q its charge in units of elementary charge, and k a constant whose value depends on the units used to express the magnetic field and the particle momentum $\vec{p}$. The constant k equals 0.3 if the field and momenta are expressed in Tesla and GeV/c, respectively. Given the magnitude of the velocity v is a constant, the Lorentz factor $\gamma = (1 - v^2/c^2)^{-1/2}$ is also a constant, and Eq. (9.12) may be rewritten

$$\frac{d^2\vec{r}}{d^2l} = \frac{kq}{p}\frac{d\vec{r}}{dl} \times \vec{B}(\vec{r}(l)), \tag{9.13}$$

where l is the path length of the track and $dl/dt = v$. Choosing $\vec{B}$ to be parallel to the z-axis, a solution of this equation for uniform $\vec{B}$ corresponds to a helicoidal trajectory with an axis parallel to z:

$$\begin{aligned} x &= x_0 + R\left[\cos\left(\Phi_0 + \frac{hl\sin\Theta}{R}\right) - \cos\Phi_0\right], \\ y &= y_0 + R\left[\sin\left(\Phi_0 + \frac{hl\sin\Theta}{R}\right) - \sin\Phi_0\right], \\ z &= z_0 + l\cos\Theta, \end{aligned} \tag{9.14}$$

where $\vec{r}_0 = (x_0, y_0, z_0)$ is the starting point of the helix at $l = 0$, and $\Theta = \arcsin(\partial z/\partial l)$ determines the **pitch** of the helix. The radius of the helix is $R = p\sin\Theta/|kqB|$ while $h = -\text{sign}(qB_z)$ determines the direction of rotation of the helix in the transverse plane. The angle Φ_0 corresponds to the azimuth angle of the starting point in cylindrical coordinates with respect to the axis of the helix, as illustrated in Figure 9.14. The helix trajectory is thus completely specified by only five parameters: x_0, y_0, z_0, R, and Φ_0. This choice is not unique. Many other combinations of five parameters may also be used to uniquely define a helix. For instance, one could have alternatively chosen the center of the helix in the transverse plane ($x_c = x_0 - R\cos\Phi_0, y_c = y_0 - R\sin\Phi_0$), the radius R, and the helix pitch λ. Other choices are also possible.

There indeed is much freedom in the choice of the five track parameters used to express the state of particle $\vec{s}$ at a given detection surface. In practice, the choice of a specific set of parameters is usually determined by convenience and the geometry of the tracking detector. The product of the track radius and azimuthal angle of hits, $R\Phi$, is a convenient parameter choice for cylindrical detector layers, but Cartesian coordinates are more frequently used because most tracking devices involve a multilayered planar detector geometry. The preceding parameterization, Eq. (9.14), while convenient for tracking in a uniform magnetic field is somewhat cumbersome for experiments where the field varies through the apparatus. A more detailed discussion of track parameterizations may be found in the book by Frühwirth et al. [89].

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

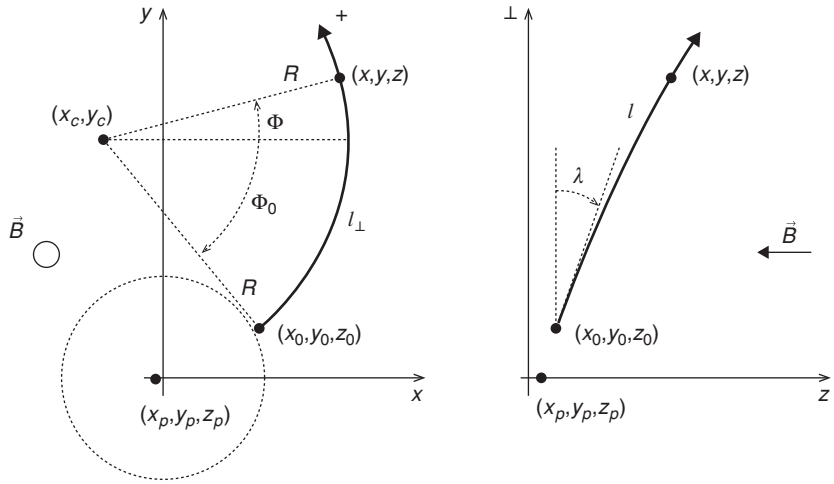


Fig. 9.14 Schematic illustration of a basic helicoidal track parameterization: R and (x_0, y_0, z_0) are the radius and first point measured on the helix; (x, y, z) represents an arbitrary point along the helix. (x_p, y_p, z_p) and (x_c, y_c) are the primary vertex and the geometrical center of the helix in the transverse plane $\perp$, respectively.

Track Propagation Model

As illustrated in Figure 9.15, a **track propagation model** or simply **track model** $\vec{f}_{j|i}(\vec{s}_i)$ is required to describe how the track parameters, $\vec{s}$, or **state vector**, can be propagated from one detector surface i to a different surface j.

$$\vec{s}_j = \vec{f}_{j|i}(\vec{s}_i) \tag{9.15}$$

An analytical track model can usually be formulated for simple detection surfaces (e.g., planar and cylindrical surfaces) operated in a vanishing (straight lines) or homogeneous magnetic field (helicoidal trajectories). However, propagation of trajectories in inhomogeneous magnetic fields and nonelementary surfaces requires the use of numerical methods such as Runge–Kutta integration techniques.

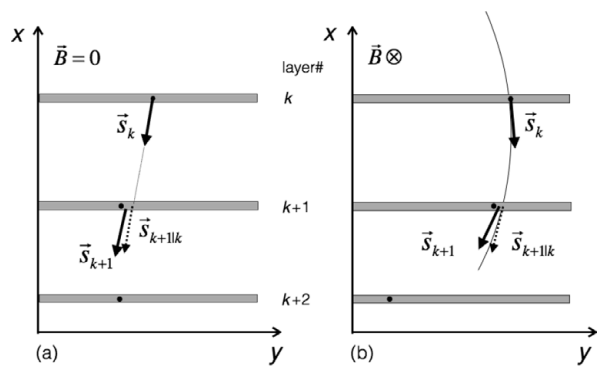


Fig. 9.15 Schematic illustration of a track propagation models for (a) $\vec{B} = 0$ and (b) finite + uniform field.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

Error Propagation

Propagation of the track parameter covariance matrix, $\mathbf{S}_i$, along with its state $\vec{s_i}$, is required in Kalman fit and adaptive fit methods. This can be usually accomplished using linear error propagation techniques (see §2.11). For instance, given the error covariance matrix at the detection surface i, one may extrapolate the state of the track to surface j with Eq. (9.15), and the extrapolated covariance matrix can then be written

$$\mathbf{S}_{j|i} = \mathbf{F}_{j|i}\mathbf{S}_i(\mathbf{F}_{j|i})^T \tag{9.16}$$

where the transformation matrix $\mathbf{F}_{j|i}$ is defined according to

$$\mathbf{F}_{j|i} = \frac{\partial \vec{s_j}}{\partial \vec{s_i}} \tag{9.17}$$

The functions $\mathbf{F}_{j|i}$ are analytical for analytical track models, but numerical techniques or semianalytical techniques must be used when propagation through inhomogeneous fields is involved [60].

Effect of Detector Materials

Charged particles propagating through a detector undergo interactions with materials present along their path. Such interactions are typically dominated by ionization energy loss and multiple Coulomb scattering. Energy loss by emission of Bremsstrahlung radiation also plays an important role for light particles such as electrons. In well-designed detectors, fluctuations in energy loss are relatively small and can be treated as deterministic corrections to the state vector. Fluctuations associated with Bremsstrahlung radiation are typically large and thus affect both the state vector and its covariance matrix. Multiple coulomb scattering is an elastic and stochastic process. With thin or low density scatterers, it only disturbs the direction of motion of charged particles, but with thick scatters one must also account for a change in position in the plane of scattering materials. However, since the expectation value of the scattering process is null, only the covariance matrix of the state vector needs updating. An example of treatment of such effects is presented in §9.2.3.

Measurement Model

The measurement model, denoted $\vec{h}(\vec{s})$, expresses the functional dependence of a measurement vector, $\vec{m}_k$, in a specific detector layer k on the state vector $\vec{s_k}$ of a particle in that same layer:

$$\vec{m}_k = \vec{h}(\vec{s_k}) \tag{9.18}$$

Much flexibility is available in the definition of measurement vector $\vec{m}$. While position measurements (y_k, z_k) in the detection plane k are most commonly used, other choices are possible that may involve measurements of direction or even the momentum of the particle.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

It is convenient to define a projection matrix $\mathbf{P}_k$ as

$$\mathbf{P}_k = \frac{\partial \vec{m}_k}{\partial \vec{s}_k}. \tag{9.19}$$

This projection matrix is typically needed for state and covariance matrix propagation which account for measurement errors. The follow-your-nose algorithm uses the track parameters at layer k to carry out a projection to layer $k-1$ (or other inner layers) to obtain an estimate of the position $(x_{k-1}, y_{k-1}, z_{k-1})$ of the "next" hit. A search is thus conducted for matching hits in the vicinity of this point with a maximum search radius r_{k-1}. The search radius may be kept constant throughout the extension (i.e., for all layers) or may be tailored to progressively shrink as the length of the track increases. This is most naturally accomplished in the context of a Kalman filter algorithm, which provides layer-by-layer, estimates of the track parameter covariance matrix. If two or more matching hits are found within the search radius, one may elect to use the closest, or attempt track extensions with all matching candidates and choose the one that eventually leads to the longest and highest quality track. The extension of a track stops once no more hits can be found or when the inner-most layer is reached. The track can be optionally extended outward toward other detectors, if any. It is eventually saved/stored if it meets required minimal quality criteria. The track reconstruction then switches to seed finding for a new track. Once all hits have been used or no more tracks are found, one can optionally initiate a second pass with looser search criteria.

Once track reconstruction is completed, one can then proceed to combine all tracks and identify the position of the primary vertex. When the collision vertex is identified, an additional pass is typically performed on all tracks to attempt a projection to this vertex and carry out a track refit that includes the vertex. Since the primary vertex is the origin of many tracks, particularly in heavy-ion collisions, it is possible to achieve very high precision in the determination of its position. A track refit including this very precise position thus enables an improved determination of the momentum and direction of primary tracks. It is also possible to carry out a refit that excludes outlier hits based on some preset maximum fit residue Δr.

A fit of the track and projection at the primary vertex yields its (transverse) momentum and direction at the point of point of production. In collider geometries, the direction is usually specified in terms of an azimuthal angle and a pitch angle or pseudorapidity. The use of polar angles measured to the beam axis tends to be reserved for fixed-target experiments or forward-going particles in collider geometries.

Track Seed Finding

The principle of seed finding algorithms is shown in Figure 9.16, where we use the variable x to express the position of tracking planes, and the variables y (and z) to identify the position of hits in a given plane.

Track reconstruction is often initiated in a region of low hit density, typically the detection planes farthest from the production vertex, and proceeds "inward" toward regions of higher hit and track density. The algorithm proceeds iteratively: a seed is identified, and

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

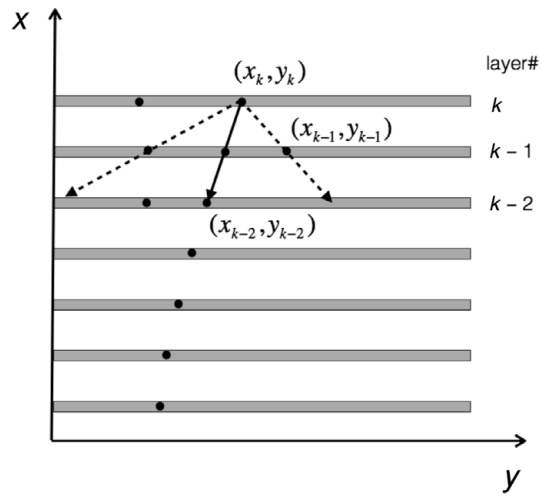


Fig. 9.16 Illustration of the linear seed-finding technique.

it is progressively extended using the method described in the text that follows. Once the innermost track layer is reached or preset number of tracking planes are traversed without successful addition of hits, the track may be fitted and stored if it meets minimal quality criteria or otherwise simply abandoned and deleted. Most tracking algorithms limit the use of hits to one track per hit, that is, they do not allow hit sharing between tracks. Hits associated with a track are thus flagged as "taken" and removed from the pool of hits to be associated with tracks. However, track reconstruction with hit sharing is also possible.

Track seeds are formed by associating hits from two to five (rarely more) contiguous outer tracking layers, as illustrated in Figure 9.13. Beginning, say, on the outermost layer, $k = n$ (where n is the number of tracking planes), one selects a hit and search on layer $k - 1$ for matching hits. In order to limit random associations, hits belonging to neighboring layers are usually required to be within a preset maximal distance along the y- and z-axes. One may also begin the search with nearly "vertical" tracks by setting the y–z ranges to be very narrow, and recursively increase the search radius up to its maximum preset value. Once a second hit is found, one proceeds to calculate the parameters of straight lines in the x–y and x–z planes formed by these two hits according to

$$\begin{aligned} a_{1,y} &= \frac{y_k - y_{k-1}}{x_k - x_{k-1}} \qquad a_{0,y} = y_k - a_{1,y} x_k, \\ a_{1,z} &= \frac{z_k - z_{k-1}}{x_k - x_{k-1}} \qquad a_{0,z} = z_k - a_{1,z} x_k \end{aligned} \quad . \tag{9.20}$$

The track parameters are used to extrapolate the tracklet to layer $k - 2$ according to

$$y_{k-2} = a_{0,y} + a_{1,y} x_{k-2}, \tag{9.21}$$

$$z_{k-2} = a_{0,z} + a_{1,z} x_{k-2}. \tag{9.22}$$

Then, one searches for hits in the vicinity of the point (y_{k-2}, z_{k-2}) within a preset maximum search radius. If the search is not successful, one may optionally attempt to find matching

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

hits on layer $k-3$ or $k-4$. If a matching hit found, it is added to the tracklet. The search can be optionally extended to additional inner layers. If the tracking planes used to form the seed are located in a field-free region, one proceeds with a simple linear fit to obtain straight line parameters in the x–y and x–z planes. If the planes are within a finite magnetic field, one carries a fit using a helicoidal track model. Either way, one gets an initial estimate of the track parameters. The tracklet may then be passed to the main tracking routine for an extended search through the rest of the detector.

9.2.3 Kalman Filter Track Finding and Fitting

Most track-fitting techniques are based on linear least-squares methods. The linear, global least-square method is optimal [178] if the track model itself is linear, that is, if the propagation function defined by Eq. (9.15) is a linear function of the state vector $\vec{s}$. In cases where the functions $\vec{f}(\vec{s})$ are nonlinear, the linear least-squares method remains the optimal linear estimator but may lack robustness [166]. Least-squares track-fitting methods are discussed in a generic context in ref. [178]. Here, we focus our discussion on an example of the Kalman filter method largely developed for track and vertex finding by Billoir [42] and Frühwirth [86].[2] The principle and properties of Kalman filters (KFs) were already discussed in §5.6, where we showed, in particular, that KFs are strictly equivalent to least-squares methods. KF methods also present the advantage, particularly in the context of track finding and fitting, that only small number of matrix inversions are required to carry out track fits, while global fit methods may require repeated inversion of large matrices [178].

As an example of application of the Kalman filter, we consider the reconstruction of charged particle tracks within a uniform solenoid magnetic field spectrometer, as schematically illustrated in Figure 8.15. The detection system consists of n_s identical azimuthal sectors each divided into n_p measurement planes. Collisions take place near the origin of the coordinate system and produce particles streaming out at high momentum. Charged particles are deflected by the magnetic Lorentz force and follow helicoidal trajectories one describes according to Eq. (9.14).

The coordinates x_c, y_c represent the center of the circle formed by projecting the helicoidal trajectory in the x–y plane traverse to the beam–beam collision axis, which defines the z-axis. R, φ, and λ are the radius, the phase angle, and the pitch of the helix, as defined in Figure 9.14. We arbitrarily choose the x-axis to be local to each sector and along a radius normal to the measurement planes. The y-axis is chosen along the measurement planes and perpendicular to the beam direction. In this context, the position of the detection planes along the x-axis constitutes a natural choice for the independent variable t. Helicoidal trajectories (9.14) involve nine variables and three independent equations. Choosing x as the independent variable leaves $(9-3-1)=5$ independent track state parameters. The choice of these five parameters is somewhat arbitrary. Given that position measurements in each detection plane are performed in the y–z plane, it is natural to use

[2] The example presented is based on the software developed by the author for the STAR experiment at the Relativistic Heavy-Ion Collider.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

these two coordinates as part of the Kalman state. The measurement vector is set to be

$$\vec{m}_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix}, \tag{9.23}$$

where y_k and z_k are the coordinates of track hits on measurement plane $k = 1, \ldots, n_p$. We similarly choose those two coordinates to be the first two elements of the Kalman state vector. This leaves us with only three more parameters to define or identify. Many choices are still possible. For illustrative purposes, we choose variables η, C, and ξ. C is the curvature defined as the multiplicative inverse of the radius R of the track,

$$C \equiv \frac{1}{R}. \tag{9.24}$$

The variable ξ is defined as the tangent of the pitch angle of the track which yields

$$\xi = \tan\lambda = \frac{p_z}{p_T}, \tag{9.25}$$

where p_T and p_z are the transverse and longitudinal momentum of the charged particle that produced the track, respectively. Defining the azimuthal angle φ of the track as

$$\tan\varphi = \frac{p_y}{p_x}, \tag{9.26}$$

one calculates η according to

$$\eta = Cx - \sin\varphi. \tag{9.27}$$

The track Kalman momentum state on plane k, denoted $\vec{s}_k$, is thus

$$\vec{s}_k = \begin{bmatrix} y_k \\ z_k \\ \eta_k \\ C_k \\ \xi_k \end{bmatrix}. \tag{9.28}$$

With the preceding choices for $\vec{m}_k$ and $\vec{s}_k$, the measurement matrix, $\mathbf{H}_k$, is reduced to a trivial form:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{9.29}$$

Inserting this specific form for $\mathbf{H}_k$ in Eq. (5.164), we get

$$K_k = \mathbf{S}_{k|k-1} (H_k)^T \tag{9.30}$$
$$\times \left(\mathbf{V}_k + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The propagation (projection) of the state from plane k to $k+1$ is accomplished by incrementing the independent variable x from x_k to $x_{k+1} = x_k + \Delta x_{k,k+1}$, where $\Delta x_{k,k+1}$ is

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

the distance between the two detector planes. The projected state vector is written

$$\vec{s}_{k+1} = \mathbf{F}\left(\vec{s}_k\right). \tag{9.31}$$

Since the state projection involves a displacement of the position of the track in a uniform longitudinal magnetic field, the parameters η, C, and ξ remain unchanged:

$$\eta_{k+1} = \eta_k, \tag{9.32}$$

$$C_{k+1} = C_k, \tag{9.33}$$

$$\xi_{k+1} = \xi_k. \tag{9.34}$$

Only the positions y and z need updating. Using the geometry illustrated in Figure 9.14, one finds after some simple algebra

$$y_{k+1} = f_1(\vec{s}_k) = y_k + \frac{\sin\varphi_k + \sin\varphi_{k+1}}{\cos\varphi_k + \cos\varphi_{k+1}}, \tag{9.35}$$

$$\begin{aligned} z_{k+1} &= f_2(\vec{s}_k) = z_k + \xi_k\left(\varphi_{k+1} - \varphi_k\right)/C_k, \\ &= m_k + \frac{\xi_k}{C_k}\sin^{-1}\left(\frac{\left(\sin\varphi_{k+1} - \sin\varphi_k\right)\left(\sin\varphi_{k+1} + \sin\varphi_k\right)}{\sin(\varphi_k + \varphi_{k+1})}\right), \end{aligned} \tag{9.36}$$

where the sine functions can be expressed in terms of η according to Eq. (9.27).

The propagation of the state error matrix is accomplished with

$$\mathbf{S}_{k|k-1} = \mathbf{F}_k\mathbf{S}_{k-1|k-1}(\mathbf{F}_k)^T + \mathbf{W}_k, \tag{9.37}$$

where $\mathbf{S}_{k-1|k-1} \equiv \mathbf{S}_{k-1}$ is the state error matrix obtained after the $(k-1)$th step of Kalman filtering. The propagation of $\vec{s}$ determined by Eq. (9.31) is nonlinear and makes the propagation of the error matrix rather cumbersome. Fortunately, since the propagation of errors does not have to be as precise as the propagation of the state itself, one can then use a linearization of Eq. (9.31) to obtain the projection matrix $\mathbf{F}_k$ required in Eq. 9.37. Consider the error at step k is of order $\Delta\vec{s}_k$. The error at step $k+1$ may then be written

$$\Delta\vec{s}_{k+1} = \mathbf{F}\left(\vec{s}_k + \Delta\vec{s}_k\right) - \mathbf{F}\left(\vec{s}_k\right). \tag{9.38}$$

Using a truncated Taylor expansion of the first term on the right-hand side, one gets

$$\Delta\vec{s}_{k+1,j} = \sum_{j=1}^{5}\Delta\vec{s}_{k,j}\frac{\partial f_j(\vec{s}_k)}{\partial s_j}. \tag{9.39}$$

The covariance matrix $\mathbf{S}_{k+1}$ is defined as the expectation value of $\langle\Delta\vec{s}_{k+1,i}\Delta\vec{s}_{k+1,j}\rangle$ over an ensemble of measurements at step $k+1$. This can be written

$$\langle\Delta\vec{s}_{k+1,i}\Delta\vec{s}_{k+1,j}\rangle = \sum_{m,n=1}^{5}\frac{\partial f_i(\vec{s}_k)}{\partial s_m}\frac{\partial f_j(\vec{s}_k)}{\partial s_n}\langle\Delta\vec{s}_{k,i}\Delta\vec{s}_{k,j}\rangle. \tag{9.40}$$

We thus obtain the matrix $\mathbf{F}$ as

$$(\mathbf{F}_k)_{i,m} = \frac{\partial f_i(\vec{s}_k)}{\partial s_m}. \tag{9.41}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

In order to calculate the preceding derivatives, we first note

$$\frac{\partial \sin\varphi}{\partial \eta} = \frac{\partial (Cx - \eta)}{\partial \eta} = -1, \tag{9.42}$$

$$\frac{\partial \cos\varphi}{\partial \eta} = \tan\varphi, \tag{9.43}$$

$$\frac{\partial \sin\varphi}{\partial C} = x, \tag{9.44}$$

$$\frac{\partial \cos\varphi}{\partial C} = -x\tan\varphi. \tag{9.45}$$

We next define $u = \sin\varphi_{k+1}\cos\varphi_k - \cos\varphi_{k+1}\sin\varphi_k$ and compute the following derivatives:

$$\frac{\partial z_{k+1}}{\partial u} = \frac{\xi}{C\sqrt{1-u^2}}, \tag{9.46}$$

$$\frac{\partial z_{k+1}}{\partial \sin\varphi_{k+1}} = \frac{\xi}{C\sqrt{1-u^2}}\cos\varphi_k, \tag{9.47}$$

$$\frac{\partial z_{k+1}}{\partial \sin\varphi_k} = -\frac{\xi}{C\sqrt{1-u^2}}\cos\varphi_{k+1}, \tag{9.48}$$

$$\frac{\partial z_{k+1}}{\partial \cos\varphi_k} = \frac{\xi}{C\sqrt{1-u^2}}\sin\varphi_{k+1}, \tag{9.49}$$

$$\frac{\partial z_{k+1}}{\partial \cos\varphi_{k+1}} = -\frac{\xi}{C\sqrt{1-u^2}}\sin\varphi_k. \tag{9.50}$$

Derivatives of z_{k+1} with η and C can then be written

$$\begin{aligned}\frac{\partial z_{k+1}}{\partial \eta} = {} & \frac{\xi}{C\sqrt{1-u^2}}\left(\xi\sin\varphi_{k+1} - \xi\sin\varphi_k\right. \\ & \left. + \xi\cos\varphi_{k+1} - \xi\cos\varphi_k\right),\end{aligned} \tag{9.51}$$

$$\begin{aligned}\frac{\partial z_{k+1}}{\partial C} = {} & \frac{\xi}{C\sqrt{1-u^2}} \times \left(-x_1\xi\sin\varphi_{k+1} + x_2\xi\sin\varphi_k\right. \\ & \left. - x_1\xi\cos\varphi_{k+1} + x_2\xi\cos\varphi_k\right).\end{aligned} \tag{9.52}$$

Based on these, we can next calculate the elements of the matrix $\mathbf{F}$. All diagonal elements F_{ii} are equal to unity. Other non-null elements are as follows:

$$F_{13} = \Delta x_{k+1,k}\frac{-2\left(\cos\varphi_k + \cos\varphi_{k+1}\right) - \left(\sin\varphi_k + \sin\varphi_{k+1}\right)\left(\tan\varphi_k + \tan\varphi_{k+1}\right)}{\left(\cos\varphi_k + \cos\varphi_{k+1}\right)^2},$$

$$F_{14} = \Delta x_{k+1,k}\frac{\left(x_k + x_{k+1}\right)\left(\cos\varphi_k + \cos\varphi_{k+1}\right) + \left(\sin\varphi_k + \sin\varphi_{k+1}\right)\left(\tan\varphi_k + \tan\varphi_{k+1}\right)}{\left(\cos\varphi_k + \cos\varphi_{k+1}\right)^2},$$

$$F_{23} = \frac{\xi}{C\sqrt{1-u^2}}\left(\xi\sin\varphi_{k+1} - \xi\sin\varphi_k + \cos\varphi_{k+1} - \cos\varphi_k\right),$$

$$F_{24} = \frac{\xi}{C\sqrt{1-u^2}}\left(-x_k\xi\sin\varphi_{k+1} + x_{k+1}\xi\sin\varphi_k - x_k\cos\varphi_{k+1} + x_{k+1}\cos\varphi_k\right),$$

$$F_{25} = \Delta x_{k+1,k}\frac{\sin\varphi_k + \sin\varphi_{k+1}}{\sin(\varphi_k + \varphi_{k+1})}.$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

An energy loss correction may also be included in the propagation of the tracks through the different detection planes. Propagating tracks inward, that is, toward the production vertex, one must add the average loss of tracks traversing the detection plane. The energy of the track is calculated according to

$$E_{k+1} = E_k + \Delta E_{k,k+1} = E_k + \Delta x_{k,k+1} \left| dE/dx \right| , \tag{9.53}$$

where dE/dx is calculated with the Bethe–Bloch formula, Eq. (8.131). The added energy implies an increase in momentum, which translates into a change of curvature:

$$C_{k+1}(E_{k+1}) = C_k(E_k) + \Delta E_{k,k+1} \left. \frac{dC}{dE} \right|_{E_k} . \tag{9.54}$$

The radius of curvature of the track, $R = 1/C$, is proportional to its momentum. One gets

$$\frac{dC}{dE} = -C \frac{\sqrt{p^2c^2 + m^2c^4}}{p^2}, \tag{9.55}$$

which yields the following correction for the curvature

$$C_{k+1}(E_{k+1}) \approx C_k(E_k) \left[1 - \frac{\sqrt{p^2c^2 + m^2c^4}}{p^2} \Delta E \right] . \tag{9.56}$$

Given the explicit dependence on the (unknown) mass of the particle, one must carry the correction with an arbitrary choice of particle identification, for instance, the mass of the pion, which is abundantly produced in high-energy collisions. However, the choice of mass used in Eq. (9.56) has typically a relatively small impact on the reconstruction of the charged particle tracks with very high momenta.

Multiple coulomb scattering through various material layers of the detector contributes a process noise to the propagation of the state covariance matrix (§9.5). In the local reference frame of a track, the scatterings cause tracks to be deflected vertically and horizontally by angles we label as γ_1 and γ_2, as illustrated in Figure 9.17.

For a thin scatterer, the noise process covariance matrix can then be estimated with the expression

$$\mathbf{W} = \langle \gamma^2 \rangle \left(\frac{\partial(\eta, C, \tan\lambda)}{\partial(\gamma_1, \gamma_2)} \right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \left(\frac{\partial(\eta, C, \tan\lambda)}{\partial(\gamma_1, \gamma_2)} \right)^T , \tag{9.57}$$

which after considerable algebra reduces to

$$\mathbf{W}_k = \langle \gamma^2 \rangle \begin{bmatrix} x^2C^2\xi^2 + (1+\xi^2)\cos^2\varphi & x^2C^2\xi^2 & xC\xi(1+\xi^2) \\ xC^2\xi^2 & C^2\xi^2 & C\xi(1+\xi^2) \\ xC\xi(1+\xi^2) & C\xi(1+\xi^2) & (1+\xi^2) \end{bmatrix} , \tag{9.58}$$

with

$$\langle \gamma^2 \rangle = \left(\frac{14.1}{p\beta} \right)^2 \frac{X}{X_o}, \tag{9.59}$$

where X and X_o stand for the thickness and the radiation length of the scatterer, respectively.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

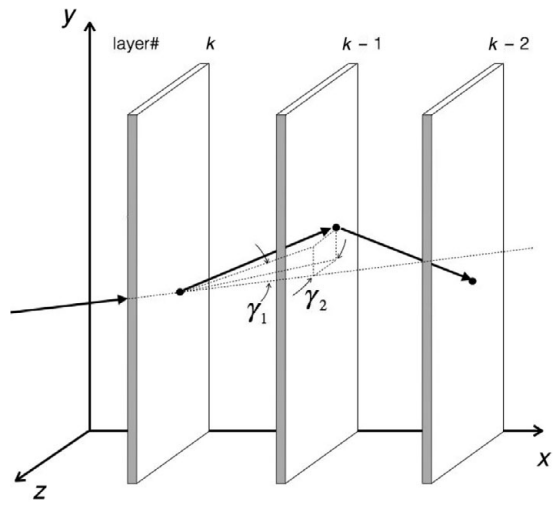


Fig. 9.17 Track geometry and multiple coulomb scattering. The scattering angles γ_1 and γ_2 have been exaggerated for the sake of clarity.

9.2.4 Removal of Hit Outliers and Competition for Hits

Track candidates produced by a track-finder can involve one or more spurious hits that belong to another track or are the result of detector noise. Given that such hits may steer or distort a track, it is appropriate to consider a third track reconstruction stage, known as **outlier removal**, where one proceeds to eliminate outlier hits, that is, hits that appear to lie too far from the track obtained by a fit.

Rejection of hit outliers is in principle readily accomplished using a χ^2 cut. Indeed, if the hit errors are known and well calibrated, the χ^2 of a fit including an outlier point is expected to be large. One can, for instance, compare the χ^2 obtained with and without a hit under consideration. If the χ^2 that includes the hit exceeds a preset value (relative to the one that excludes it), the point is simply removed from the track. One may equivalently consider the normalized fit residue of the hit based on a fit that excludes it. However, the presence of several outliers may bias or skew a track so much that they render either of these procedures completely inadequate.

An alternative approach is to increase the robustness of the fit, thereby reducing the influence of potential outliers. This can be accomplished using adaptive estimation techniques, discussed in the next section, which automatically reduce the weight of outlying observations. Yet another technique involves the use of Tukey's bi-square function (see. e.g., [95, 101]).

Another common issue arising in track finding and fitting is that two tracks may have one or more hits in common. This is particularly the case in track pairs produced by the Hanbury-Brown and Twiss (HBT) effect: two particles of same charge sign may be emitted with nearly the same momenta and angles and thus propagate near one another through the detector. Finite hit resolution may then cause two tracks to share one or several hits.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

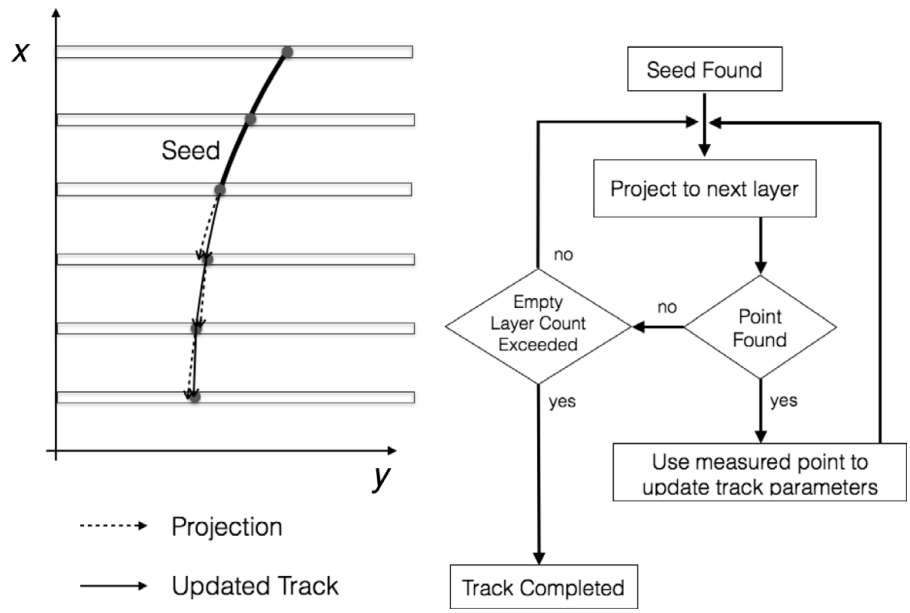


Fig. 9.18 Example of progressive track recognition with a Kalman filter algorithm. A seed formed in the outer layers of the detector is extended inwards through a recursive process involving projection of the track to the next inner layer and use of a matching point to update the track parameters.

Such tracks may be considered incompatible or at the very least biased, since shared hits are likely to be merged hits with suboptimal accuracy because the actual two hits they correspond to cannot be resolved. A technique based on graphs in which every track corresponds to a node is presented in Das [77]. Such technique may result in the production of several solutions of the same size (i.e., number of tracks). It may then be desirable to choose tracks also based on their quality. This is accomplished most generally by assigning each track with a quality index (based on its χ^2, length, distance of closest approach to the primary vertex, etc.) The best node is the one that maximizes the sum of quality indices. A technique to find the best node based on Hopfield networks is discussed in [178] and references therein.

9.2.5 Hybrid Track Finding and Fitting Techniques

The Kalman filter method can conveniently be used to achieve track finding and track-fitting concurrently [40, 43] with the algorithm illustrated in Figure 9.18. The process starts with a track seed obtained from hits located in contiguous layers. The seed is extended iteratively to adjacent layers based on the track parameters determined by the Kalman fit. A track candidate is abandoned if too few compatible hits are found and added to the track.

The χ^2 of the residual $\vec{r}_{k|k-1}$ of the measurement $\vec{m}_k$ with respect to the predicted state can be used as an indicator of the compatibility of a given hit to a track:

$$\chi^2_{k,+} = \vec{r}^{\,T}_{k|k-1} \mathbf{V}^{-1}_{k|k-1} \vec{r}_{k|k-1}, \tag{9.60}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

where

$$\vec{r}_{k|k-1} = \vec{m}_k - \vec{h}_k(\vec{s}_{k|k-1}), \tag{9.61}$$

and the covariance matrix of the residuals is

$$\mathbf{V}_{k|k-1} = \mathbf{V}_k + \mathbf{H}_k \mathbf{S}_{k|k-1} \mathbf{H}_k^T. \tag{9.62}$$

In cases where several hits on a given detector layer might be compatible with a track projection, one should choose the one that yields the lowest χ^2, given by Eq. (9.60), add it to the track candidate, update the Kalman state, and use it to extend (predict) the track further.

A generalized Kalman filter technique, known as **combinatorial Kalman filter**, may be used to deal with high track density environments or a high density of noisy measurements [139]. Global/local hybrid track finding techniques based on cellular automata have also been developed to cope with such situations [94, 127]. A generic Kalman finder/fitter named GenFit is reported in [108].

9.3 Primary Vertex Reconstruction Techniques

A vertex consists of a point where particles are produced either as a result of a collision between beam and target particles, by the decay of a particle, or by an interaction of a particle with the material of the detector. The vertex corresponding to a beam + target leads to the production of **primary particles** and is known as a **primary vertex**, whereas vertices corresponding to decays are known as **secondary vertices**. While there are many commonalities between the identification of primary and secondary vertices, we focus in this section on the reconstruction of primary vertices.

The primary vertex of a collision corresponds to the point (location) where the collision between the beam and target particles took place. In fixed-target experiments, the position of primary vertices is largely constrained by the transverse cross section of the beam and the physical thickness of the target. By contrast, in collider experiments the position of the vertex may vary considerably from collision to collision owing to the longitudinal extension of the colliding beam bunches. The volume spanned by collisions is commonly known as the **collision diamond**. The longitudinal and transverse sizes of the collision diamond are determined by the beam crossing angle and the bunches' size, two parameters essentially fixed by the design of a collider. While the transverse size of the diamond is typically rather small (e.g., few hundred microns to few millimeters), the longitudinal extension of the diamond may be rather long. For instance, at RHIC the Au – Au diamond initially had an root mean square of approximately 30 cm. This implies one cannot rely on the collision longitudinal diamond geometry, and one must determine the primary vertex position collision by collision (event by event). Knowledge of the position of the primary vertex of a collision is required first and foremost in order to be able to distinguish primary, secondary, and tertiary particles. It is also useful to establish cuts that suppress pile-up events

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

(i.e., corresponding to two or multiple overlapping collisions), useful particularly in high-luminosity environments such as the LHC proton–proton running. The primary vertex is typically also used in the refit of primary tracks, given that, as we shall see in the text that follows, the precision of the position of the primary vertex usually far exceeds that of hits on tracks and thus enables a very precise constraint in the determination of the curvature of tracks. Precise knowledge of the primary vertex also enables, once secondary vertices are found, the measurement of decay length of short-lived particles.

Similarly to track reconstruction, the task of vertex reconstruction nominally involves two stages known as **vertex finding** and **vertex fitting**. Vertex finding typically begins with all valid tracks produced by the track reconstruction, supplied as a list of track parameter vectors. We discuss the basic algorithms and methods used for vertex finding in §9.3.1, which by and large all yield the same outcome consisting of a list of vertex candidates with their associated tracks. These candidates are then fed to a vertex fitter, which attempts to use all associated tracks to obtain an optimal vertex position. Selected vertex-fitting techniques are discussed in §9.3.2.

9.3.1 Vertex Finding

Vertex finding is a task involving the association and classification of reconstructed tracks of an event into one or several vertex candidates. Three types of vertices are nominally found. A **primary vertex** corresponds to the point of interaction between two beam particles (in a collider geometry) or a beam and a target (in a fixed-target geometry), while particles produced by the decay of an unstable particle (e.g., $\Lambda^0 \to p + \pi^-$) are said to originate from a **secondary vertex**. A secondary vertex (rarely called tertiary vertex) may also correspond to the point of production of background particles via the interaction of primary or secondary particles with the material of the detector.

Primary vertices are usually straightforward to find, particularly in heavy-ion collisions where they are associated with tens to thousands of particles. Exceptions arise in electron-positron colliders, for instance, for collisions that produce two short-lived neutral particles (e.g., $\Upsilon(4S) \to B^0\bar{B}^0$). Secondary vertices are usually limited to two or three particles and are thus more challenging to identify, particularly if they are produced in close proximity to a primary vertex in central heavy-ion collisions.

A large number of techniques have been devised and implemented in experiments to determine the position of vertices and their associated tracks. We present few illustrative examples of such methods and refer the reader to a recent comprehensive review by Strandlie for additional methods and details [178].

Track Clustering Methods

Clustering methods use the distance or proximity between objects (e.g., tracks) or measures of similarity to classify objects into clusters or groups. A cluster consists of a group of objects located within a small distance from one another (or with many shared features), but at a larger distance (or dissimilar features) relative to members of other clusters. There

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

are a number of techniques to form clusters, but given the relative nature of the notion of proximity, it is useful to think in terms of hierarchical clustering or a tree of clusters. Each object is a leaf on the tree. Objects that are closest together form a small branch, and objects of two small branches that are close to one another are associated into a bigger branch, and so on. Clustering methods that begin with the leaves are said to be agglomerative or associative, whereas those that start at the root, that is, with all the objects, are known as divisive methods.

In high-luminosity experiments (e.g., in p–p collisions at the LHC), recorded events may involve a plurality of primary vertices. In some instances, it may be possible to record and analyze all vertices and their associated tracks, but in general, selection criteria are applied to choose which vertex is actually of interest. In proton–proton collisions, this could for instance be the interaction vertex with the largest number of high transverse momentum tracks (determined based on a sum of the p_T of all the tracks associated with a vertex), while in heavy-ion experiments, one is typically interested in the collision with the largest multiplicity of associated tracks.

However, note that in TPC-based experiments such as STAR or ALICE, the long drift and readout time of the TPC may lead to the pile-up of many collisions, and the collision that triggers the recording of an event is not necessarily the one featuring the largest multiplicity. Additionally, note that at very high luminosity, there is a finite probability that two distinct collisions may have primary vertices located so close to one another as to be virtually indistinguishable.

Various hierarchical and nonhierarchical clustering methods have been evaluated and documented in the literature (see [187] and references therein). In hierarchical agglomerative clustering, each track starts out as a single cluster. Clusters are then merged iteratively on the basis of a distance measure. It is important to realize that if two tracks a and b are close, and tracks b and c are also close, this does not imply that tracks a and c are necessarily close to one another. The distance between two clusters of tracks must therefore be defined as the maximum of the individual pairwise distances, known as complete linkage in the clustering literature. In contrast, divisive clustering starts out with a single cluster containing all tracks. This and subsequent clusters are iteratively divided based on some maximum distance criterion identifying the presence of outliers. Outliers are removed from a cluster and become the seed of one or several new clusters. The procedure is iterated until all produced clusters are smaller than a certain preset size and have no outlier tracks.

The **histogramming method** (also known as the **mapping method**) is an agglomerative technique based on hits from (pixel) detector layers located in close proximity to the beam axis (or nominal interaction region). It was implemented by the CMS experiment [71] and various other prior experiments but is otherwise little discussed in the literature, perhaps because of its intrinsic simplicity of design and implementation.

The technique consists in histogramming the point of intersection of tracklets, m_{ip}, (track consisting of three hits) with the nominal beam axis. Empty bins are discarded and nonempty bins are scanned. A cluster is defined as a contiguous set of bins separated by a distance smaller than a preset upper limit $\Delta m_{\max}$. Clusters are ranked according to their number of associated tracklets, and only clusters with a number of tracklets exceeding a

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

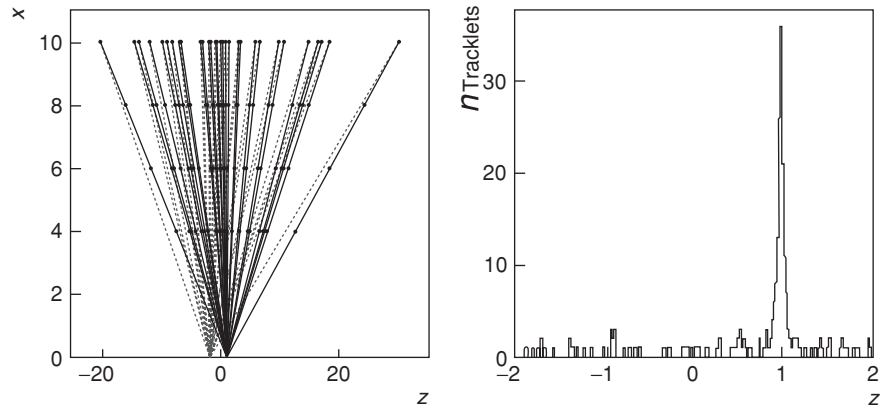


Fig. 9.19 Illustration of the principle of the vertex histogramming method. Fifty tracks were generated by Monte Carlo to originate from a vertex position at $z = 1$ cm and to produce hits on four detector layers as shown on the left. To find this vertex, one scanned the z-axis in small steps, as explained in the text, to find the position that yields the largest number of tracklets, as shown on the right, where one recovers the actual position of the vertex as a peak in the number of tracklets reconstructed at $z = 1$.

preset threshold are considered. Their positions and estimated errors are calculated based on a weighted average.

A variant of the technique, shown in Figure 9.19, consists in scanning the z-vertex axis in small steps. For each step, one determines the pitch angle, $\lambda = \tan^{-1}(z/x)$, of the hits and fills a histogram with these angles. High p_T tracks produce hits with similar pitch angle in the detector layers near the beam axis. One then counts the number of bins with more than three hits and fills a histogram with the number of tracklets with three or more hits. Primary vertices are those positions with a number of tracklets exceeding a preset threshold. This variant presents the advantage of being extremely fast since the number of operations scales as the number of hits while the first histogramming technique discussed earlier scales roughly as the number of hits to the 3rd (or 4th) power for 3 (4) hit tracklets.

Other approaches are also possible. For instance, a divisive clusterizer based on the intersection of tracklets, m_{in}, with the z-axis has been implemented by the CMS experiment [71]. Tracks are sorted by increasing value of m_{in} and the ordered list is scanned. Clusters are initiated and terminated when the gap between two consecutive track intersections exceeds a nominal threshold. For each initial cluster, an iterative procedure is applied to remove incompatible tracks. The discarded tracks are recovered to form a new cluster, and the same procedure is applied iteratively until there are fewer than two remaining tracks.

Linear Finders

Linear finders use charged particle track extrapolations near the beam axis (collider geometry) and local linear approximations. Rather than seeking a position P that minimizes the (average) distance of closest approach to all primary charged tracks, it is simpler and much

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

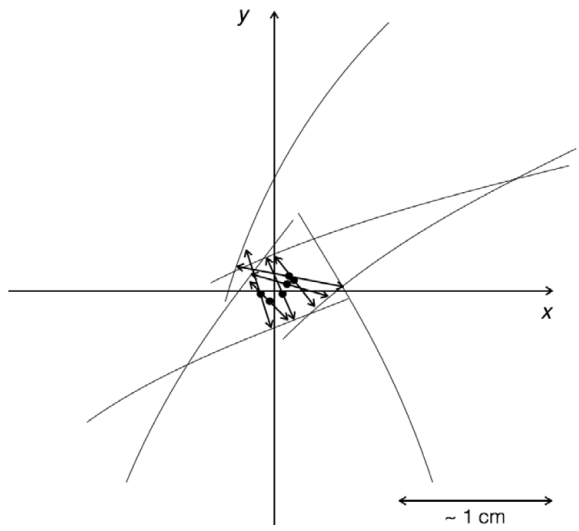


Fig. 9.20 Determination of the primary vertex position based on track pair DCAs. The vertex position is obtained as the three-dimensional average of all track pair DCA midpoint positions. Only a handful of tracks and pair midpoints are shown for the sake of clarity.

faster to calculate the mean position of the DCA midpoint of all primary track pairs, as illustrated in Figure 9.20.

The DCA midpoint, $\vec{r}_{ij}$ of a pair of tracks (i and j), corresponds to the bisection point of a segment $\Delta\vec{r}$ joining points at closest approach. A technique to calculate the position of the DCA midpoint is presented in §9.4. The position of the primary vertex $\vec{S}_v$ is estimated as an average of the DCA midpoint position of all pairs of charged tracks:

$$\hat{S}_v = \frac{2}{n(n-1)} \sum_{i,j=1}^{n} \vec{r}_{ij}, \tag{9.63}$$

where n is the number of tracks considered in the estimation of the vertex position. Given an event typically contains a combination of primary, secondary, and tertiary tracks, and possibly several vertices, one must ensure that the position determined with all tracks is not arbitrarily pulled toward specific secondary or tertiary particles. This can be accomplished in part by effecting a maximum DCA cut on the pairs included in the sum. One can also use an iterative procedure to eliminate outlier tracks from the sample used to evaluate the primary vertex position. Once an estimate of the primary vertex position is obtained, one calculates the DCA of all tracks of the sample (using the track-to-point DCA technique described in §9.4) to eliminate tracks that feature a DCA to the primary vertex in excess of a selected cut value. The vertex position can then be recalculated using the estimator, Eq. (9.63), using only tracks that satisfy the DCA cut. The procedure can be iterated until no further tracks are removed or too few tracks are left in the sample. This method is found to produce usable and reliable primary vertices when the number of tracks exceeds three.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

Topological Vertex Finders

A general topological vertex finder method, known as ZVTOP, was introduced by Jackson [112]. It is related to the Radon transform, which is a continuous version of the Hough transform used for track finding (§9.2.1). The method and its recent developments are discussed in [178].

9.3.2 Vertex Fitting

Vertex fitting methods commonly used in particle and nuclear physics experiments include least-squares fitters, notably the Billoir vertex fitter [45], Kalman filter based fitters, as well as Kalman filters extended into adaptive vertex fitters. The Billoir vertex fitter uses all tracks associated to the vertex at once and estimates the vertex position by means of a global least-squares fit. By contrast, Kalman vertex fitters use a seed and proceed iteratively to add tracks to the vertex. Adaptive vertex fitters are an extension of the Kalman filter method and assign tracks incrementally to a vertex according to their compatibility with the vertex. A detailed description of Kalman filter vertex fitters and adaptive vertex fitters is well beyond the scope of this textbook, but the reader is referred to the review by [178] for details on these methods. For illustrative purposes, we here describe the Billoir vertex fitter following the original development of the method by its authors [45].

The Billoir vertex fitting method is based on a local parameterization of tracks in the vicinity of the vertex point. For convenience, one defines two different coordinate systems: a global coordinate system defined as the interaction point, and a local coordinate system in the neighborhood of the vertex point. After the vertex fit, this origin moves to the vertex point. The goal is to estimate the vertex position in the global reference frame and use it to recalculate the parameters of tracks in the local (vertex) frame. We demonstrate the applicability of the method for a uniform magnetic field of intensity $|\vec{B}|$ and the helix model stated in Eq. (9.14) as a starting point. The basic idea of the method is to project the helix trajectories near the local origin of the coordinate system. One may then use a first-order approximation to describe the local curvature of the tracks and employ all tracks to obtain an estimate of the vertex position by minimizing a χ^2 function. In so doing, one will obtain an "optimal" vertex position and updated track states and momentum vectors $\vec{p}_i$. The parameters relevant for the determination of the DCA to the z-axis are illustrated in Figure 9.21, in which d_0 is the transverse impact parameter of the projected helix. It corresponds to the DCA, or perigee, of the helix to the z-axis. m_0 is defined as the longitudinal impact parameter of the helix and corresponds to the z position of the particle at perigee. ϕ_0 is the azimuth angle of the momentum vector at the perigee. It is by convention measured in the range $[-\pi, \pi]$; θ is the polar angle, measured in the range $[0, \pi]$; and q/p the ratio of the charge to the momentum of the particle. The distance d_0 is signed according to the convention illustrated in Figure 9.21. It is defined as positive if $\phi - \phi_0 > 0$ and negative otherwise. In the vicinity of a vertex, the trajectory $\vec{r}$ of the particle may be locally expressed as

$$\vec{r}(l) \approx \vec{r}(l_0) + (l - l_0) \left.\frac{d\vec{r}}{dl}\right|_{l_0} + \frac{(l - l_0)^2}{2} \left.\frac{d^2\vec{r}}{dl^2}\right|_{l_0} \tag{9.64}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

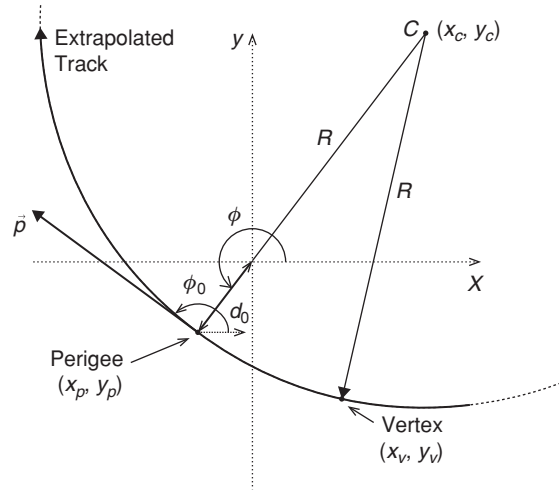


Fig. 9.21 Track perigee parameters used in the determination of production vertices with the Billoir vertex fitting method [45].

The parameters d_0, m_0, ϕ, θ, and q/p of the point of closest approach to the local z-axis are shown in Appendix §9.4.3. One finds

$$d_0 = R_0 + \frac{L^2}{2hR}, \tag{9.65}$$

$$z_0 = z_P - L\cot\theta_P, \tag{9.66}$$

$$\phi = \phi_P - \frac{L}{hR}, \tag{9.67}$$

$$\theta = \theta_P, \tag{9.68}$$

$$\frac{q}{p} = \frac{q}{p_P}. \tag{9.69}$$

Given prior estimates of the vertex position $\vec{V}_0$, measured track states $\vec{s}_{i,0}$ and momenta $\vec{p}_{i,0}$, we wish to find new momenta $\vec{p}_i = \vec{p}_{i,0} + \delta\vec{p}_i$ that correspond to the "real" vertex position $\vec{V} = \vec{V}_0 + \delta\vec{V}$. We then seek to minimize a χ^2 function defined as

$$\chi^2 = \sum_{i=1}^{n} \Delta\vec{s}_i^{\,T} \mathbf{W}_i \Delta\vec{s}_i, \tag{9.70}$$

where n is the number of tracks nominally associated with the vertex and $\Delta\vec{s}_i = \vec{s}_{i,0} - \vec{F}(\vec{V}, \vec{s}_i)$. The function $\vec{F}$ determines the state a track should have given a vertex position $\vec{V}$ and momentum vector $\vec{p}$. By minimizing the preceding χ^2 function, one finds the changes $\delta\vec{V}$ and $\delta\vec{s}_i$ that are most compatible with the measured states $\vec{s}_{i,0}$. In principle, the function $\vec{F}$ has a nonlinear dependency on both $\delta\vec{V}$ and $\delta\vec{s}_i$. However, if the initial estimate $\vec{V}_0$ is reasonably close to the real value $\vec{V}$, it is sufficient to use a linear approximation of the function $\vec{F}$, and one writes

$$\vec{F}\left(\vec{V}, \vec{s}_i\right) \simeq \vec{F}\left(\vec{V}_0, \vec{s}_{i,0}\right) + D_i\delta\vec{V} + E_i\delta\vec{s}_i, \tag{9.71}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

where the coefficients D_i and E_i are derivatives of $F(\vec{V}, \vec{s_i})$ with respect to $\vec{V}$ and $\vec{s_i}$ evaluated at $\vec{V}_0$ and $\vec{s}_{i,0}$. Substituting the preceding expression in Eq. (9.70), one gets the quadratic function

$$\chi^2 \simeq \sum_{i=1}^{n} \left(\delta\vec{s_i} - D_i\delta\vec{V} - E_i\delta\vec{p_i}\right)^T \mathbf{W}_i \left(\delta\vec{s_i} - D_i\delta\vec{V} - E_i\delta\vec{p_i}\right), \tag{9.72}$$

where $\delta\vec{s_i} = \vec{s}_{i,0} - \vec{F}(\vec{V}_0, \vec{s}_{i,0})$. Minimization of the χ^2 must be accomplished simultaneously relative to $\vec{V}$ and all $\vec{p_i}$. Setting derivatives with respect to these quantities equal to zero, one gets

$$0 = \sum_{i=1}^{n} \left(D_i^T W_i D_i \delta\vec{V} + D_i^T W_i E_i \delta\vec{p_i} - D_i^T W_i \delta\vec{s_i}\right), \tag{9.73}$$

$$0 = E_i^T W_i D_i \delta\vec{V} + E_i^T W_i E_i \delta\vec{p_i} - E_i^T W_i \delta\vec{s_i}. \tag{9.74}$$

For convenience, in order to solve for $\delta\vec{V}$ and $\delta\vec{p_i}$, let us define the following coefficients and vectors

$$A = \sum_i D_i^T W_i D_i, \tag{9.75}$$

$$B_i = D_i^T W_i E_i, \tag{9.76}$$

$$C_i = E_i^T W_i E_i, \tag{9.77}$$

$$\vec{T} = \sum_i D_i^T W_i \delta\vec{s_i}, \tag{9.78}$$

$$\vec{U_i} = E_i^T W_i \delta\vec{s_i}, \tag{9.79}$$

where the sums are taken over particles $i = 1, \ldots, n$. Equations (9.73) and (9.74) may then be written

$$0 = A\delta\vec{V} + \sum_i B_i \delta\vec{p_i} - \vec{T}, \tag{9.80}$$

$$0 = B_i^T \delta\vec{V} + C_i \delta\vec{p_i} - \vec{U_i}. \tag{9.81}$$

Solving for $\delta\vec{V}$ and $\delta\vec{p_i}$, one obtains the updated vertex and momenta according to

$$\begin{aligned} \vec{V} &= \vec{V}_0 + \delta\vec{V} \\ &= \vec{V}_0 + \left(A - \sum_i B_i C_i^{-1} B_i^T\right)^{-1} \left[\vec{T} - \sum_i B_i C_i^{-1} \vec{U_i}\right], \end{aligned} \tag{9.82}$$

$$\begin{aligned} \vec{p_i} &= \vec{p}_{i,0} + \delta\vec{p_i} \\ &= \vec{p}_{i,0} + C_i^{-1}\left(\vec{U_i} - B_i^T \delta\vec{V}\right). \end{aligned} \tag{9.83}$$

Covariance matrices Cov$[\vec{V}, \vec{V}]$, Cov$[\vec{p_i}, \vec{p_i}]$, and Cov$[\vec{V}, \vec{p_i}]$ are provided in [45].

Equations (9.82,9.83) together constitute a $(3n + 3) \times (3n + 3)$ matrix equation that may be rather tedious solve, particularly if the number of tracks n is very large. However, Billoir and Qian developed a perigee parameterization that alleviates the inversion of

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

such a large matrix and uses only small matrices. Discussions of this and more advanced techniques based on Kalman and adaptive Kalman filter methods are beyond the scope of this work but may be found in the recent literature [44, 45, 71, 88, 112, 178, 187].

9.3.3 Performance Characterization

A primary vertex finder can be characterized on the basis of its efficiency, the associated track multiplicity, and the position resolution of the vertices it produces.

The vertex-finding efficiency, ϵ_V, is defined as the ratio of the number of vertices actually found, n_{found}, divided by the actual number of vertices, n_{produced}.

$$\epsilon_V = \frac{n_{\text{found}}}{n_{\text{produced}}} \tag{9.84}$$

The vertex-finding efficiency typically depends on a wide variety of experimental factors, including the experimental trigger, the track reconstruction efficiency and resolution, as well as the specific technique used to reconstruct vertices. While the reconstruction efficiency may be roughly estimated based on the data, it is generally obtained via careful Monte Carlo modeling (simulations) of the experimental apparatus and its performance.

The number of tracks associated with a particular vertex, or track multiplicity, provides a basic selection criterion of the quality of reconstructed vertices. It can be used to rank multiple primary vertices found in events recorded in very high luminosity environments such as the proton–proton beams at the LHC. It can additionally provide a selection criterion for reaction types of interest. A maximum multiplicity can be applied, for instance, to identify quasi diffractive collisions in proton–proton collisions or grazing heavy-ion collisions involving a single or multiple photon exchanges.

The vertex position resolution improves as the number of pairs included in a measurement of the vertex position increases. Let δr_{ij} represent the resolution of the midpoint positions $\vec{r}_{ij}$. Assuming the track sample consists of n primary tracks exclusively, and neglecting correlations in measurements of the positions, one estimates the resolution on $\hat{S}_v$ as

$$\delta S_v = \sqrt{\frac{2}{n(n-1)} \sum_{i,j=1}^{n} \delta r_{ij}^2}. \tag{9.85}$$

9.4 Appendix 1: DCA Calculations

9.4.1 DCA Straight Line to Fixed Point

The DCA between a straight line l_1 and a fixed point P_2 can be obtained using the geometrical construction illustrated in Figure 9.22a.

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

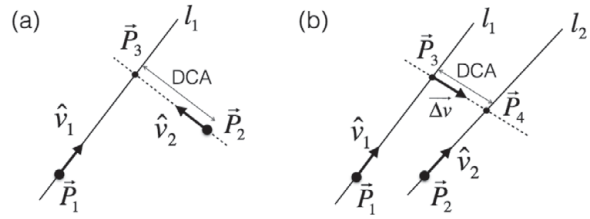


Fig. 9.22 Calculation of the DCAs between (a) a line l_1 and a fixed point S_1 and between (b) two lines l_1 and l_2.

The line l_1 is determined by a point $\vec{P}_1$ and a direction $\hat{v}_1$ (with $\hat{v}_1 \cdot \hat{v}_1 = 1$) and can thus be parameterized as

$$\vec{l_1} = \vec{P_1} + t\hat{v}_1, \tag{9.86}$$

where $t \in \mathbb{R}$. The DCA is achieved for a point $\vec{P}_3$ which belongs to l_1 and lies along a line l_2 determined by

$$\vec{l_2} = \vec{P_2} + s\hat{v}_2, \tag{9.87}$$

with $s \in \mathbb{R}$, $\hat{v}_1 \cdot \hat{v}_2 = 0$ and $\hat{v}_2 \cdot \hat{v}_2 = 1$. One thus seeks the values t and s such that $\vec{l_1} = \vec{l_2}$:

$$\vec{P_1} + t\hat{v}_1 = \vec{P_2} + s\hat{v}_2. \tag{9.88}$$

Multiplying both sides of this equation by $\hat{v}_1$, one gets

$$\vec{P_1} \cdot \hat{v}_1 + t = \vec{P_2} \cdot \hat{v}_1, \tag{9.89}$$

where we used the fact that $\hat{v}_1$ is a unit vector by construction perpendicular to $\hat{v}_2$. Solving for t, we get

$$t = \left(\vec{P_2} - \vec{P_1}\right) \cdot \hat{v}_1, \tag{9.90}$$

which uniquely determines the position of the point $\vec{P}_3$ of l_1 closest to P_2.

$$\vec{P_3} = \vec{P_1} + \left[\left(\vec{P_2} - \vec{P_1}\right) \cdot \hat{v}_1\right]\hat{v}_1. \tag{9.91}$$

The *DCA* is then

$$DCA = \left[\left(\vec{P_3} - \vec{P_2}\right) \cdot \left(\vec{P_3} - \vec{P_2}\right)\right]^{1/2} \tag{9.92}$$

9.4.2 DCA between Two Straight Lines

We calculate the DCA between two straight line l_1 and l_2 using the geometrical construction illustrated in Figure 9.22b. The two lines are determined by the points P_1 and P_2 as well as the unit vectors $\hat{v}_1$ and $\hat{v}_2$. The points P_3 and P_4 corresponding to the DCA between the two lines may be written

$$\vec{P_3} = \vec{P_1} + t_1\hat{v}_1 \tag{9.93}$$

$$\vec{P_4} = \vec{P_2} + t_2\hat{v}_2, \tag{9.94}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

where $t_1, t_2 \in \mathbb{R}$. The segment joining these two points is expressed as

$$\vec{P}_3 - \vec{P}_4 = d\Delta\hat{v}, \tag{9.95}$$

where $d \in \mathbb{R}$ is the DCA between the two lines and $\Delta\hat{v}$ is a unit vector. Substituting the expressions (9.93) for $\vec{P}_3$ and (9.94) for $\vec{P}_4$, we get

$$\vec{P}_1 + t_1\hat{v}_1 - \vec{P}_2 - t_2\hat{v}_2 = d\Delta\hat{v}, \tag{9.96}$$

which we in turn multiply by $\hat{v}_1$ and $\hat{v}_2$:

$$\vec{P}_1 \cdot \hat{v}_1 + t_1 - \vec{P}_2 \cdot \hat{v}_1 - t_2\hat{v}_1 \cdot \hat{v}_2 = 0, \tag{9.97}$$

$$\vec{P}_1 \cdot \hat{v}_2 + t_1\hat{v}_1 \cdot \hat{v}_2 - \vec{P}_2 \cdot \hat{v}_2 - t_2 = 0, \tag{9.98}$$

where we have use the fact that $\Delta\hat{v}$ is by construction perpendicular to both $\hat{v}_1$ and $\hat{v}_2$. A solution for t_1 and t_2 may be written

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} 1 & -\hat{v}_1 \cdot \hat{v}_2 \\ \hat{v}_1 \cdot \hat{v}_2 & -1 \end{bmatrix}^{-1} \begin{bmatrix} (\vec{P}_1 - \vec{P}_2) \cdot \hat{v}_1 \\ (\vec{P}_1 - \vec{P}_2) \cdot \hat{v}_2 \end{bmatrix}, \tag{9.99}$$

which determines the position of $\vec{P}_3$ and $\vec{P}_4$. The DCA is then

$$DCA = \left[\left(\vec{P}_3 - \vec{P}_4\right) \cdot \left(\vec{P}_3 - \vec{P}_4\right) \right]^{1/2}. \tag{9.100}$$

9.4.3 DCA between a Helix and the Local *z*-Axis

Based on Figure 9.21, the position of the perigee may be written

$$x \simeq -d_0 \sin\phi_P + (l \sin\theta_P)\cos\phi_P + \frac{l^2 \sin^2\theta_P}{2hR} \sin\phi_P, \tag{9.101}$$

$$y \simeq -d_0 \cos\phi_P + (l \sin\theta_P)\sin\phi_P + \frac{l^2 \sin^2\theta_P}{2hR} \cos\phi_P, \tag{9.102}$$

$$z \simeq z_P + l\cos\theta_P. \tag{9.103}$$

It is convenient to define $L = l \sin\theta_P$, which can also be written as

$$L = x\cos\phi + y\sin\phi. \tag{9.104}$$

Writing the radius R of the helix as

$$R_0 = y\cos\phi_P - x\sin\phi_P, \tag{9.105}$$

one obtains

$$R_0 = d_0 - \frac{L^2}{2hR}. \tag{9.106}$$

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press

The parameters of the perigee may then be written

$$d_0 = R_0 + \frac{L^2}{2hR}, \tag{9.107}$$

$$z_0 = z_P - L\cot\theta_P, \tag{9.108}$$

$$\phi = \phi_P - \frac{L}{hR}, \tag{9.109}$$

$$\theta = \theta_P, \tag{9.110}$$

$$\frac{q}{p} = \frac{q}{p_P}, \tag{9.111}$$

where d_0 and m_0 correspond to the DCA of the track relative to the estimated vertex.

9.5 Appendix 2: Multiple Coulomb Scattering

Charged particles passing through a detector medium undergo multiple coulomb scattering and are thus very likely to change direction. For sufficiently thick media (but not too thick), the scattering angle θ has a probability density that can be well approximated with a Gaussian distribution with an RMS angle $\theta_{\text{plane}}^{\text{rms}}$ given [35] by

$$\theta_{\text{plane}}^{\text{rms}} = \theta_0 = \frac{13.6\text{MeV}}{\beta cp} z\sqrt{\frac{x}{X_0}}\left[1 + 0.038\ln\left(\frac{x}{X_0}\right)\right], \tag{9.112}$$

where x is the thickness of the material and X_0 is the radiation length of the material. The radiation lengths of selected materials often used in the construction of detectors are listed in [35].

A well-designed detector should be thick enough to produce a signal that can be detected with high efficiency but thin enough to limit multiple coulomb scattering effects. Optimization of the detector is contingent on many factors and depends, in particular, on the signal-to-noise ratio.

Exercises

9.1 Show that the expression (5.213) simplifies to **I**.

9.2 Show that the conformal mapping defined by Eqs. (9.8) and (9.9) yields a straight line (9.11) for circular trajectories given by Eq. (9.10).

https://doi.org/10.1017/9781108241922.012 Published online by Cambridge University Press