Track and vertex reconstruction: From classical to adaptive methods

Are Strandlie*

Gjøvik University College, P.O. Box 191, N-2802 Gjøvik, Norway

Rudolf Frühwirth†

Institute of High Energy Physics of the Austrian Academy of Sciences, Nikolsdorfer Gasse 18, A-1050 Wien, Austria

(Published 7 May 2010)

This paper reviews classical and adaptive methods of track and vertex reconstruction in particle physics experiments. Adaptive methods have been developed to meet the experimental challenges at high-energy colliders, in particular, the CERN Large Hadron Collider. They can be characterized by the obliteration of the traditional boundaries between pattern recognition and statistical estimation, by the competition between different hypotheses about what constitutes a track or a vertex, and by a high level of flexibility and robustness achieved with a minimum of assumptions about the data. The theoretical background of some of the adaptive methods is described, and it is shown that there is a close connection between the two main branches of adaptive methods: neural networks and deformable templates, on the one hand, and robust stochastic filters with annealing, on the other hand. As both classical and adaptive methods of track and vertex reconstruction presuppose precise knowledge of the positions of the sensitive detector elements, the paper includes an overview of detector alignment methods and a survey of the alignment strategies employed by past and current experiments.

DOI: 10.1103/RevModPhys.82.1419 PACS number(s): 02.70.Rr, 07.05.Kf, 07.05.Mh, 29.85.Fj

CONTENTS

*Also at Department of Physics, University of Oslo, Norway. are.strandlie@hig.no

†Also at Department of Statistics and Probability Theory, University of Technology, Vienna, Austria. fru@hephy.oeaw.ac.at

I. INTRODUCTION

Significant developments have taken place during the past decades in the fields of high-energy physics accelerators, detectors, and computing technologies. The desire to understand in more detail the basic behavior of the fundamental constituents of nature and the interactions between these constituents has led to experimental studies carried out at ever increasing energies. In general, the number of particles created by the interaction of two particles rises with the collision energy. As a consequence, event patterns have become much more complex in the course of time. In addition, data rates have increased in order to enable the search for rare interesting collision events immersed in a huge background of events exhibiting well-known physics. Whereas most events in the experiments during the 1950s were studied manually on a scanning table, the data rate at the Large Hadron Collider (LHC) at CERN, Geneva, Switzerland, has to be reduced online by more than five orders of magnitude before the information from an event is written on mass storage for further analysis. Tracking detectors have changed from bubble chambers to gaseous and solid-state electronic detectors. Computers have changed from large mainframes to server farms with processor speeds several orders of magnitude higher than those available some decades ago. The task of analyzing data from high-energy physics experiments has therefore, over the years, been performed in a continuously changing environment, and analysis methods have evolved accordingly to adapt to these changes.

This paper aims to review methods employed in crucial parts of the data-analysis chain in a high-energy physics experiment: the task of determining the basic kinematic parameters of charged particles at their point of production and the task of estimating the location of these production points. These tasks are frequently called track reconstruction and vertex reconstruction. Track and vertex reconstruction are done both online, especially in the high-level trigger, and offline. Online applications frequently use simplified methods for the sake of speed. The following discussion will concentrate on methods employed in the offline analysis, where the ultimate precision is the major goal and speed is rarely a decisive factor.

Track and vertex reconstructions can attain, however, their ultimate precision only if the positions and orientations of the sensitive detector elements are known to high accuracy. The review therefore includes detector alignment methods that are used for the precise determination of the alignment constants (positions and orientations) of the sensitive detector elements. In the experiments at the LHC the number of detector elements to be aligned runs into several thousands and in one case even exceeds 10 000. Furthermore, the required precision is considerably higher than the intrinsic resolution of the detector elements which in itself is already very small, on the order of 10 μm. Detector alignment is thus a highly nontrivial task that can ultimately only be solved using information from charged particles that cross several sensitive elements along their trajectory through the detector. Alignment is therefore intimately connected to the track reconstruction challenge.

The topics of track and vertex reconstruction have, over the years, been reviewed several times with focus ranging from bubble-chamber data-analysis methods (Jobes and Shaylor, 1972) through approaches applied in early electronic experiments (Eichinger and Regler, 1981; Grote, 1987) up to methods used in currently running experiments (Regler *et al.*, 1996; Frühwirth *et al.*, 2000; Mankel, 2004). In this review, however, we put less emphasis on traditional approaches but rather focus on more recently developed adaptive methods tailored to the needs of experiments of the LHC era. We also make an attempt to describe the theoretical background of some of these adaptive methods and show how different classes of algorithms are intimately connected to each other. The underlying theory of how charged particles interact with matter, as well as the basic detection principles and basic detector types, is excellently described in textbooks [e.g., Leo (1994), Bock and Vasilescu (1998), and Amsler *et al.* (2008)] and will not be covered here.

In contrast to a traditional approach, a method can be regarded as adaptive if it incorporates competition between several hypotheses such that the outcome of the competition depends on the current observations. In addition, the possibilities of making soft decisions and of combining hypotheses according to their posterior weights are typical features of an adaptive method. Some of the adaptive methods have a strong Bayesian flavor, reflected, for example, in the presence of posterior probabilities or weights attached to the different hypotheses. The numerical values of the weights in general 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. The hallmark of a good adaptive method is that it achieves maximum flexibility and robustness with as few assumptions about the data as possible.

The review begins in Sec. II with a description of classical methods of track and vertex reconstruction, followed in Sec. III by an overview of adaptive methods applied to the same tasks. Detector alignment is addressed in Sec. IV, and the conclusions are given in Sec. V. An outlook to future research containing our opinions about the most important unsolved problems in the topics described in this article is given in Sec. VI.

II. CLASSICAL METHODS OF TRACK AND VERTEX RECONSTRUCTION

In any analysis of the data of a high-energy physics experiment, it is of crucial importance to estimate as accurately as possible the kinetic parameters of particles produced in a collision event, for example, the position, direction, and momentum of the particles at their production points. For this purpose, a set of detecting devices providing high-precision position measurements is positioned close to the beam collision area. Charged particles created in the collisions ionize the material of detecting devices on their way out of the collision area, providing several position measurements along the trajectory of each particle. The detector elements should disturb the trajectory of the particles as little as possible. Hence, the amount of material present in such tracking detectors should be kept at a minimum.

The task of track reconstruction is traditionally divided into two different subtasks: track finding and track fitting. Track finding is a pattern recognition or classification problem and aims at dividing the set of measurements in a tracking detector into subsets, each subset containing measurements believed to originate from the same particle. These subsets are called track candidates. An additional subset contains measurements believed not to come from any of the interesting tracks but, for instance, from noise in the electronics or from low-energy particles spiraling inside the tracking detector. Track finding should be conservative and keep a track candidate in case of doubt rather than discarding it, as a track candidate discarded at this stage is impossible to recover at any later stage. An example of a hard track finding problem is shown in Fig. 1. It is the task of the track finding to reconstruct the correct classification of the hits, shown in the bottom panel, to their respective tracks, shown in the top panel.

The track fit takes the set of measurements in a track candidate as a starting point. The goal is to estimate as accurately as possible a set of parameters describing the state of the particle somewhere in the tracking detector, often at a reference surface close to the particle beam. With few exceptions [see, e.g., James (1983) and Chernov *et al.* (1993)], the estimation is based on least-squares methods. The track fit should be as computationally fast as possible, it should be robust against mistakes made during the track finding procedure, and it should be numerically stable.

The track fit is also used to decide whether the track candidate hypothesis is valid. Such a test can be based on the value of the χ^2 statistic, i.e., the sum of the squared standardized differences between the measured positions in the track candidate and the estimated positions of the track at the points of intersection of the detector devices. If the value of such a statistic is too high, the set of measurements is not statistically compatible with the hypothesis of having been created by a single particle. The reason for this incompatibility could be a single or a few measurements in a track candidate misclassified by the track finding, or a track candidate being completely wrong in the sense that it is a random collection of measurements originating from several other particles—a so-called ghost track. The track fit should, in this testing phase, be able to remove wrong or outlying measurements in the track candidates and suppress the ghost tracks completely.

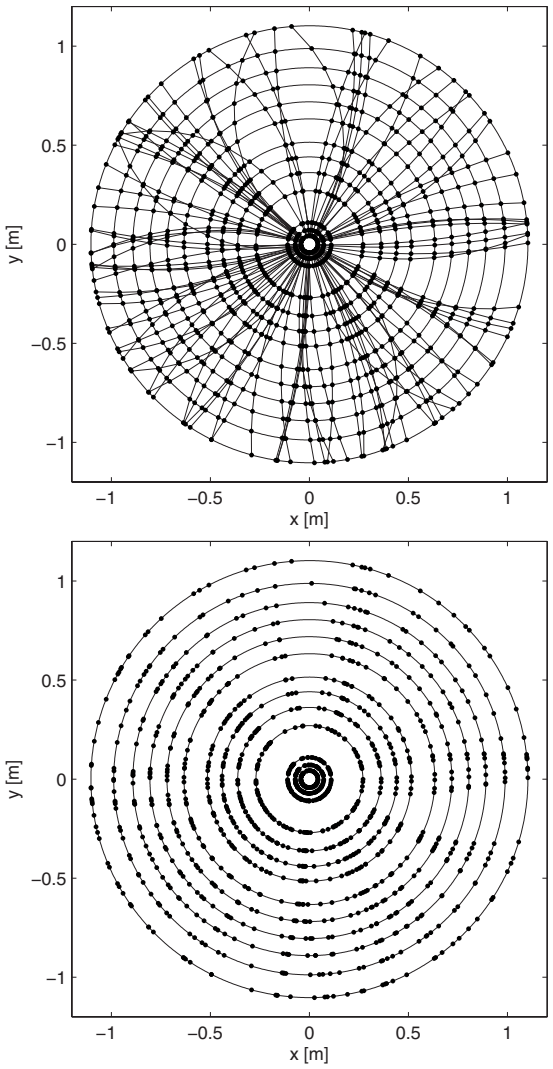

FIG. 1. A hard track finding problem. Top: 100 tracks in a cylindrical tracking detector with 13 layers. The tracks and the position measurements (hits) are shown in the projection transverse to the magnetic field. In this projection the track model is a circle. Bottom: Only the hits are shown.

The step following track reconstruction is vertex reconstruction. A vertex is a point where particles are produced either by a collision of a beam particle with another beam particle or a target particle, the decay of a particle, or by an interaction of a particle with the material of the detector. Vertex reconstruction offers the following benefits.

- Using the vertex constraint, the momentum estimates of the particles involved can be improved.
- A neutral or very short-lived particle can be reconstructed by finding its decay products and fitting them to a common vertex.

- The decay length of a short-lived particle can be determined by computing the distance between its estimated production vertex and its estimated decay vertex.

Similar to track reconstruction, the task of vertex reconstruction can be divided into vertex finding and vertex fitting. The starting point of vertex finding is the set of all valid tracks provided by the track reconstruction, represented by a list of track parameter vectors. The vertex finding algorithms classifies the tracks into vertex candidates, which are fed into the vertex fit. The output of the vertex fit is a list of vertices, each entry containing the estimated vertex position as well as a set of updated track parameter vectors of the particles associated to that particular production point. Again the χ^2 or a related statistic can be used to test the vertex hypothesis.

A. Track finding

In experimental conditions such as those found in the LHC experiments, many of the measurements are either noise or belonging to particles with energy too low to be interesting from a physics point of view. Therefore, many hypotheses have to be explored in order to find the set of interesting track candidates, and track finding can in general be a cumbersome and time-consuming procedure. Computational speed is an important issue, and the choice of algorithms may be dictated by this fact. Track finding often uses the knowledge of how a charged particle moves inside the bulk of the detector, the so-called track model, but can resort to a simplified version if time consumption is critical. The use of simplified track models is particularly important for triggering applications, where track finding is part of the strategy applied in the online selection procedure of interesting events. Such applications are not considered in this paper, which will concentrate on methods used for offline analysis of data, i.e., analysis of data available on mass storage.

Methods of track finding can in general be classified as global or local. Global methods treat all measurements simultaneously, whereas local methods go through the list of measurements sequentially. Examples of global approaches presented below are conformal mapping, Hough transform, and Legendre transform, whereas the track road and track following methods are regarded as local.

1. Conformal mapping

The conformal mapping method (Hansroul *et al.*, 1988) for track finding is based on the fact that circles going through the origin of a two-dimensional x-y coordinate system map onto straight lines in a u-v coordinate system by the transformation

$$u = \frac{x}{x^2 + y^2}, \quad v = \frac{y}{x^2 + y^2}, \tag{1}$$

where the circles are defined by the circle equation $(x-a)^2+(y-b)^2=r^2=a^2+b^2$. The straight lines in the u-v plane are then given by

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

For large values of r or, equivalently, high-momentum tracks, the straight lines are passing close to the origin, and track candidates can be obtained by transforming the measurements in the u-v plane to azimuthal coordinates and collecting the angular part of the measurements in a histogram. Track candidates are found by searching for peaks in this histogram.

2. Hough and Legendre transforms

In the case of straight lines not necessarily passing close to the origin, i.e., for tracks in a larger range of momenta, a more general approach is needed in order to locate the lines. The Hough transform (Hough, 1959) is well suited for such a task. The idea is based on a simple transformation of the equation of a straight line in an x-y plane, $y=cx+d$, to another straight line in a c-d plane, $d=-xc+y$. The points along the line in the c-d plane correspond to all possible lines going through the point (x,y) in the x-y plane. Points lying along a straight line in the x-y plane therefore tend to create lines in the c-d plane crossing at the point which specifies the actual parameters of that line in the x-y plane. In practice, the c-d space is often discretized, allowing a set of bins to be incremented for each of the measurements in the x-y space. As for the conformal mapping method, the position of peaks in the histogram provides information about the parameters of the lines in the x-y space. In contrast to the one-dimensional parameter space of the conformal mapping method, the parameter space is in this case two dimensional. The Hough transform rapidly loses efficiency for finding tracks if one attempts to move to a parameter space with a dimension higher than 2.

For track finding in drift tubes, the drift circles provided by the knowledge of the drift distances of each of the measurements can be transformed to sine curves in the azimuthal coordinate system by applying a Legendre transform (Alexopoulos *et al.*, 2008). Peaks at the intersections of several sine curves in this coordinate system give not only the set of drift tubes hit by the same particle but also the solution to the left-right ambiguity problem inherent to this kind of detector system. An illustration is shown in Fig. 2.

3. Track road

An example of a local approach to track finding is the so-called track road method. It is initiated with a set of measurements that could have been created by the same

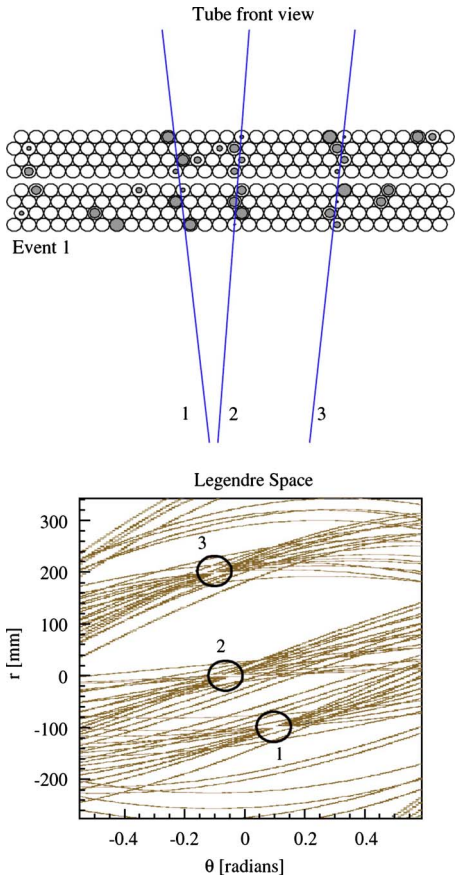

FIG. 2. (Color online) An illustration of track finding with the Legendre transform. Top: Drift chamber with a multitrack event with noise level of 50%. Bottom: The corresponding Legendre transform. The circles in Legendre space graphs denote the points with the highest height, corresponding to the reconstructed tracks shown in the top graph. From Alexopoulos *et al.*, 2008.

charged particle. The track model, i.e., the shape of the trajectory, can be used to interpolate between the measurements and create a road around the trajectory. Measurements inside the boundaries of the road constitute the track candidate. The number of measurements and the quality of the subsequent track fit are used to evaluate the correctness of the track hypothesis.

4. Track following

A related approach is track following, which starts from a track seed. Most of the times, the seed is a short track segment built from a few measurements. In addition it can be constrained to point to the interaction region. Seeds can be constructed in the inner region of the tracking detector close to the interaction region, where the measurements frequently are of very high precision, or in the outer region, where the track density is lower. From the seed, the track is extrapolated to the next detector layer containing a measurement. The measurement closest to the predicted track is included in the track candidate. This procedure is iterated until too many detector layers with missing measurements are encountered or until the end of the detector system is reached.

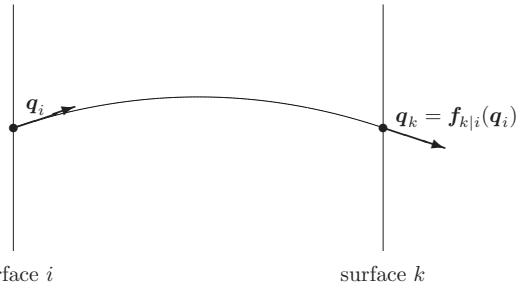

FIG. 3. An illustration of the track model and propagation concepts. The function $f_{k|i}$ is the track propagator from surface i to surface k. Its mathematical form depends on the track model, i.e., the solution of the equation of motion in the actual magnetic field.

B. Track fitting

The track fit aims at estimating a set or vector of parameters representing the kinematic state of a charged particle from the information contained in the various position measurements in the track candidate. Since these positions are stochastic quantities with uncertainties attached to them, the estimation amounts to some kind of statistical procedure. In addition to estimated values of the track parameters, the track fit also provides a measure of the uncertainty of these values in terms of the covariance matrix of the track parameter vector. Most estimation methods can be decomposed into a set of basic building blocks, and the methods differ in the logic of how these blocks are combined.

1. Track parametrization

If tied to a surface, five parameters are sufficient to uniquely describe the state of a charged particle. The actual choice of track parameters depends on, e.g., the geometry of the tracking detector. In a detector consisting of cylindrical detector layers, the reference surface is often cylindrical and makes the radius times the azimuthal angle ($R\Phi$) the natural choice of one of the position parameters. In a detector consisting of planar detector layers, however, Cartesian position coordinates are more frequently used (Frühwirth *et al.*, 2000).

2. Track model

The track model describes how the track parameter or state vector at a given surface k depends on the state vector on a different surface i,

$$\boldsymbol{q}_k = \boldsymbol{f}_{k|i}(\boldsymbol{q}_i), \tag{3}$$

where $\boldsymbol{f}_{k|i}$ is the track propagator from surface i to surface k and $\boldsymbol{q}$ is the state vector. An illustration is shown in Fig. 3. For simple surfaces, the track model is analytical in a vanishing magnetic field (straight line) or in a homogeneous field (helix). If the field is inhomogeneous,

one has to resort to numerical schemes such as the Runge-Kutta integration of the equation of motion.

3. Error propagation

During the track parameter estimation procedure, propagation of the track parameter covariance matrix along with the track parameters themselves is often requested. The standard procedure for this so-called linear error propagation is a similarity transformation between layers i and k,

$$\boldsymbol{C}_k = \boldsymbol{F}_{k|i}\boldsymbol{C}_i\boldsymbol{F}_{k|i}^{\mathrm{T}}, \tag{4}$$

where $\boldsymbol{C}$ is the covariance matrix and $\boldsymbol{F}_{k|i}$ is the Jacobian matrix of the propagation from layer i to k,

$$\boldsymbol{F}_{k|i} = \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{q}_i}. \tag{5}$$

For analytical track models the Jacobian is also analytical (Strandlie and Wittek, 2006). In inhomogeneous magnetic fields, the derivatives can be calculated by purely numerical schemes or by semianalytical propagation of the derivatives in parallel to the Runge-Kutta propagation of the track parameters (Bugge and Myrheim, 1981).

4. Material effects

The most important effects on the trajectory of charged particles caused by material present in the detector volume are ionization energy loss and multiple Coulomb scattering (Amsler *et al.*, 2008). For light particles such as electrons, radiation energy loss by bremsstrahlung also plays an important role. The fluctuations of ionization energy loss are usually quite small, and such energy loss is therefore normally treated during track fitting as a deterministic correction to the state vector (Frühwirth *et al.*, 2000). Bremsstrahlung energy loss, on the other hand, suffers from large fluctuations (Bethe and Heitler, 1934) and affects therefore both the state vector and its covariance matrix. Multiple Coulomb scattering is an elastic process, which in a thin scatterer disturbs only the direction of a passing charged particle; in a sufficiently thick scatterer, the position in a plane transversal to the incident direction is also changed (Amsler *et al.*, 2008). Since the mean value of the scattering angle and an eventual offset is zero, only the covariance matrix is updated in order to incorporate the effects of multiple scattering into the track fitting procedure.

5. Measurement model

The measurement model $\boldsymbol{h}_k$ describes the functional dependence of the measured quantities in layer k, $\boldsymbol{m}_k$, on the state vector at the same layer,

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{q}_k). \tag{6}$$

The vector of measurements $\boldsymbol{m}_k$ usually consists of the measured positions but can also contain other quantities, e.g., measurements of direction or even momentum. During the estimation procedure the Jacobian $\boldsymbol{H}_k$ of this transformation is often needed,

$$\boldsymbol{H}_k = \frac{\partial \boldsymbol{m}_k}{\partial \boldsymbol{q}_k}. \tag{7}$$

In many cases the Jacobian contains only rotations and projections and can thus be computed analytically.

a. Least-squares methods for track fitting

The overwhelming majority of experimental implementations use some kind of linear least-squares approach for the task of track fitting. The linear global least-squares method is optimal if the track model is linear, i.e., if the track propagator $\boldsymbol{f}_{k|i}$ from detector layer i to detector layer k is a linear function of the state vector $\boldsymbol{q}_i$ and if all probability densities encountered during the estimation procedure are Gaussian. If the track propagator is nonlinear, the linear least-squares method is still the optimal linear estimator. However, although least-squares estimators are easy to compute, they lack robustness (Rousseeuw and Leroy, 1987).

The starting point for deriving the global least-squares method is the functional relationship between the initial state $\boldsymbol{q}_0$ of the particle at the reference surface and the vector of measurements $\boldsymbol{m}_k$ at detector layer k,

$$\boldsymbol{m}_k = \boldsymbol{d}_k(\boldsymbol{q}_0) + \boldsymbol{\gamma}_k, \tag{8}$$

where $\boldsymbol{d}_k$ is a composition of the measurement model function $\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{q}_k)$ and the track propagator functions

$$\boldsymbol{d}_k = \boldsymbol{h}_k \circ \boldsymbol{f}_{k|k-1} \circ \cdots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}. \tag{9}$$

The term $\boldsymbol{\gamma}_k$ is stochastic and contains all multiple Coulomb scattering up to layer k as well as the measurement error of $\boldsymbol{m}_k$. A linear estimator requires a linearized track model, and for this the Jacobian $\boldsymbol{D}_k$ of $\boldsymbol{d}_k$ is needed,

$$\boldsymbol{D}_k = \boldsymbol{H}_k\boldsymbol{F}_{k|k-1}\cdots\boldsymbol{F}_{2|1}\boldsymbol{F}_{1|0}, \tag{10}$$

where $\boldsymbol{H}$ is the Jacobian of $\boldsymbol{h}$ and $\boldsymbol{F}$ is the Jacobian of $\boldsymbol{f}$.

The observations $\boldsymbol{m}_k$, the functions $\boldsymbol{d}_k$, the Jacobians $\boldsymbol{D}_k$, and the noise $\boldsymbol{\gamma}_k$ can each be arranged in a single vector or matrix,

$$\boldsymbol{m} = \begin{pmatrix} \boldsymbol{m}_1 \\ \vdots \\ \boldsymbol{m}_n \end{pmatrix}, \quad \boldsymbol{d} = \begin{pmatrix} \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_n \end{pmatrix}, \quad \boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_1 \\ \vdots \\ \boldsymbol{D}_n \end{pmatrix}, \quad \boldsymbol{\gamma} = \begin{pmatrix} \boldsymbol{\gamma}_1 \\ \vdots \\ \boldsymbol{\gamma}_n \end{pmatrix}, \tag{11}$$

where n is the total number of measurement layers. The model now becomes

$$\boldsymbol{m} = \boldsymbol{d}(\boldsymbol{q}_0) + \boldsymbol{\gamma}, \tag{12}$$

and the linearized version is

$$\boldsymbol{m} = \boldsymbol{D}\boldsymbol{q}_0 + \boldsymbol{c} + \boldsymbol{\gamma}, \tag{13}$$

where $\boldsymbol{c}$ is a constant vector. The global least-squares estimate of $\boldsymbol{q}_0$ is given by

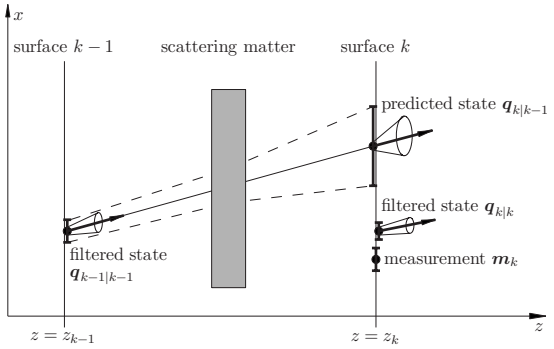

FIG. 4. Prediction and filter step of the Kalman filter. The propagation proceeds in the z direction, while the x coordinate is measured. Adapted from Regler *et al.*, 1996.

$$\tilde{\boldsymbol{q}}_0 = (\boldsymbol{D}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{D})^{-1}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{m}-\boldsymbol{c}), \tag{14}$$

where $\boldsymbol{V}=\boldsymbol{G}^{-1}$ is the nondiagonal covariance matrix of $\boldsymbol{\gamma}$.

If there is substantial multiple scattering, the estimated track can deviate significantly from the real track. The actual track can be followed more closely by explicitly estimating two projected scattering angles at each detector layer or at a set of virtual breakpoints inside a continuous scatterer (Laurikainen *et al.*, 1972; Eichinger and Regler, 1981). The breakpoint method and the global least-squares method are equivalent as far as the estimate of the state vector $\boldsymbol{q}_0$ is concerned (Billoir *et al.*, 1985).

Large numbers of measurements or breakpoints lead to a high computational cost of these methods due to the need of inverting large matrices. A recursive formulation of the least-squares method, the Kalman filter, requires the inversion of only small matrices and exhibits the same feature as the breakpoint method of following the actual track quite closely (Billoir, 1984; Frühwirth, 1987), with the advantage that material effects such as multiple scattering and energy loss can be treated locally.

The Kalman filter proceeds by alternating prediction and update steps (see Fig. 4). The prediction step propagates the estimated track parameter $\boldsymbol{q}_{k-1|k-1}$ vector from detector layer $k-1$ to the next layer containing a measurement,

$$\boldsymbol{q}_{k|k-1} = \boldsymbol{f}_{k|k-1}(\boldsymbol{q}_{k-1|k-1}), \tag{15}$$

as well as the associated covariance matrix,

$$\boldsymbol{C}_{k|k-1} = \boldsymbol{F}_{k|k-1}\boldsymbol{C}_{k-1|k-1}\boldsymbol{F}_{k|k-1}^{\mathrm{T}} + \boldsymbol{Q}_k, \tag{16}$$

where $\boldsymbol{Q}_k$ is the covariance matrix of multiple scattering after layer $k-1$ up to and including layer k. The part of $\boldsymbol{Q}_k$ arising from scattering between the layers has to be propagated to layer k by the appropriate Jacobian.

The update step corrects the predicted state vector by using the information from the measurement in layer k,

$$\boldsymbol{q}_{k|k} = \boldsymbol{q}_{k|k-1} + \boldsymbol{K}_k[\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{q}_{k|k-1})], \tag{17}$$

where the gain matrix $\boldsymbol{K}_k$ is given by

$$\boldsymbol{K}_k = \boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^{\mathrm{T}}(\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^{\mathrm{T}})^{-1}, \tag{18}$$

and $\boldsymbol{V}_k$ is the covariance matrix of $\boldsymbol{m}_k$. The covariance matrix is updated by

$$\boldsymbol{C}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{C}_{k|k-1}. \tag{19}$$

An alternative formulation of the Kalman filter operates on the inverse covariance matrices (weight or information matrices) rather than on the covariance matrices themselves (Frühwirth, 1987). This approach tends to be numerically more stable than the gain matrix formulation.

The full information at the end of the track as provided by the filter can be propagated back to all previous estimates by another iterative procedure, the Kalman smoother. A step of the smoother from layer $k+1$ to layer k is for the state vector,

$$\boldsymbol{q}_{k|n} = \boldsymbol{q}_{k|k} + \boldsymbol{A}_k(\boldsymbol{q}_{k+1|n} - \boldsymbol{q}_{k+1|k}), \tag{20}$$

where the smoother gain matrix is given by

$$\boldsymbol{A}_k = \boldsymbol{C}_{k|k}\boldsymbol{F}_{k+1|k}^{\mathrm{T}}(\boldsymbol{C}_{k+1|k})^{-1}. \tag{21}$$

The smoothed covariance matrix is

$$\boldsymbol{C}_{k|n} = \boldsymbol{C}_{k|k} - \boldsymbol{A}_k(\boldsymbol{C}_{k+1|k} - \boldsymbol{C}_{k+1|n})\boldsymbol{A}_k^{\mathrm{T}}. \tag{22}$$

The smoother can also be realized by combining two filters running in opposite directions (Frühwirth, 1987).

b. Removal of outliers and resolution of incompatibilities

A track candidate produced by the track finding algorithm can in general contain one or more outlying observations. These may be distorted hits, extraneous hits from other tracks, or electronic noise. An obvious way of rejecting outliers is to monitor the χ^2 of the observations with respect to the predicted track positions using information from all measurements but the one under consideration (Frühwirth, 1987). A cut on the χ^2 with respect to these predictions is powerful if there is only a single outlier in the track candidate. If there are several outliers, the smoothed predictions are biased, and the probability of rejecting a good observation can no longer be controlled.

Another possibility is to make the track fit more robust, thereby reducing the influence of potential outliers. The adaptive estimators presented in Sec. III.D are robust in this sense because outlying observations are automatically downweighted. A related approach can be found in Golutvin *et al.* (2000). It is based on a redescending M-estimator using Tukey's bisquare function (Hampel *et al.*, 1986).

When the track finding is completed it may happen that two-track candidates have one or more hits in common, especially if the track finding is done sequentially. Such tracks are considered as incompatible. As incompatibilities are usually forbidden, a maximal or optimal subset of compatible tracks has to be found. One way of finding such a subset is to build a graph in which every track corresponds to a node, and two nodes are connected by an edge whenever the corresponding tracks

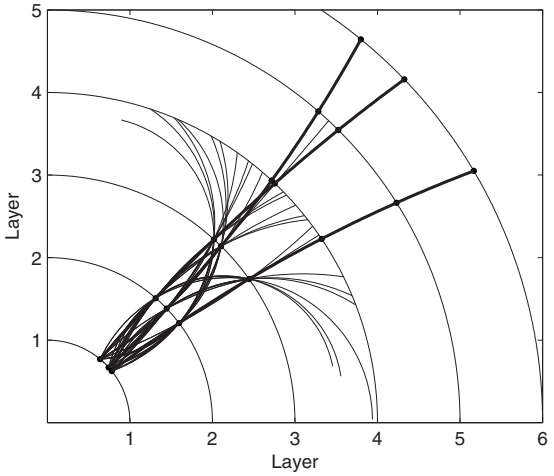

FIG. 5. Example of progressive track recognition with three tracks. Seeds are formed in layers 1–3 by finding all possible combinations of hits. Each of the seeds is extrapolated to the outer layers (thin circles). Seeds that do not reach the next layer or are not compatible with an observation in some layer are discarded. The thick circles represent the three seeds that are successfully propagated to the outermost layer.

are compatible. The problem is now to find all maximal fully connected node sets in the graph. An algorithm that solves this problem is given by Das (1973).

Searching for a maximal set of compatible tracks may give several solutions of the same size. In addition, it may be desirable to take into account the track quality. This is accomplished by assigning a quality index to each track, which can be based on various quantities such as the χ^2 statistic, the track length, the distance of the track from the interaction region, the direction of the track, etc. The best maximal compatible node set is now the one that maximizes the sum of all quality indices. Finding the best node set has been solved by a recurrent neural network (Hopfield network; see Sec. III.A). The network and an application to one of the forward chambers of the DELPHI experiment are described by Frühwirth (1993). Another algorithm, developed for the global solution of tracking ambiguities in DELPHI, is described by Wicke (1998).

c. Hybrid methods

The Kalman filter can be used for track finding and track fitting concurrently (Billoir, 1989; Billoir and Qian, 1990). The resulting progressive track finding algorithm can be regarded as an optimal track following procedure. The algorithm illustrated in Fig. 5. It starts out with finding seeds in a couple of adjacent layers, then follows each seed through the detector, and picks up compatible hits as it goes along. If not enough compatible hits are found, the candidate is dropped. In practice, some of the seeds can be discarded right away because their momentum is too small or because they do not point to the interaction point.

In the original formulation of this strategy, the χ^2 of the residual of the measurement $\boldsymbol{m}_k$ with respect to the predicted state,

$$\chi^2_{k,+} = \boldsymbol{r}^{\mathrm{T}}_{k|k-1}\boldsymbol{R}^{-1}_{k|k-1}\boldsymbol{r}_{k|k-1}, \tag{23}$$

with

$$\boldsymbol{r}_{k|k-1} = \boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{q}_{k|k-1}) \tag{24}$$

and the covariance matrix of the residual given by

$$\boldsymbol{R}_{k|k-1} = \boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}^{\mathrm{T}} \tag{25}$$

is used to evaluate the statistical compatibility of the measurement with the prediction. If there are several compatible measurements, the one with the lowest value of the χ^2 statistic is included in the track candidate and used for the update step of the Kalman filter.

If there are many nearby tracks or a high density of noisy measurements, the measurement closest to the predicted track might not necessarily belong to the track under consideration. In order to cope with such a situation, the procedure outlined above can be generalized to a so-called combinatorial Kalman filter (CKF) (Mankel, 1997). It marks the transition from classical to adaptive methods insofar as several hypotheses about the track are entertained simultaneously until in the end one of them is declared as the winner.

Like the progressive track finding, the CKF starts from a seed, usually a short track segment at the inner or the outer end of a tracking detector. If there are several compatible measurements in the first layer after the seed, several Kalman-filter branches are generated, each of them containing a unique compatible measurement at the end of the branch. In order to handle potential detector inefficiencies, a branch with a missing measurement is also created. All branches are propagated to the next detector layer containing at least one compatible measurement, and new branches are created for each combination of predicted states compatible with a measurement. This procedure leads to a combinatorial tree of Kalman filters running in parallel. Branches are removed if the total quality of the branch—in terms of the total χ^2 of the track candidate up to the layer under consideration—falls below a defined value or if too many consecutive layers without compatible measurements are traversed. In the end, the surviving branch with the highest quality—usually in terms of a combination of the total χ^2 and the total number of measurements—is kept. An example is shown in Fig. 6.

A similar track finding method has been formulated in the language of cellular automata (Glazov *et al.*, 1993; Kisel, Konotopskaya, and Kovalenko, 1997; Kisel, 2006). The cellular automaton approach can, on the one hand, be regarded as local as it builds up branches using information from measurements in nearby layers. On the other hand, the procedure is not initiated from a seed. All measurements are processed in parallel, making the approach a hybrid between a local and a global method.

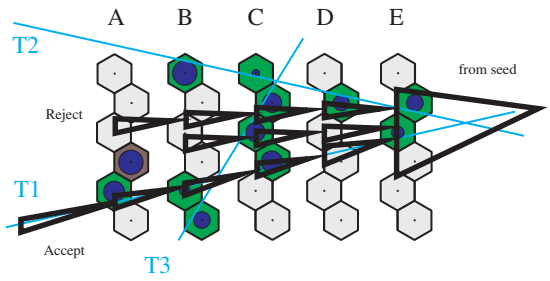

FIG. 6. (Color online) Illustration of the combinatorial Kalman filter for an ambiguous situation caused by three nearby tracks in a superlayer of the HERA-B outer tracker. The propagation proceeds upstream from the right to the left. It is assumed that the propagation started with a seed of hits from track T1. From Mankel, 1997.

C. Vertex finding

Vertex finding is the task of classifying the reconstructed tracks in an event into vertex candidates such that all tracks associated with a candidate originate at that vertex. There are several types of vertices, and different strategies may be required to find them.

- If the particles are produced by the collision of two beam particles (in a collider experiment) or a beam particle and a target particle (in a fixed-target experiment), the vertex is a primary vertex.
- If the particles are produced by the decay of an unstable particle, the vertex is a secondary decay vertex. An example is the decay $K_S^0 \to \pi^+\pi^-$.
- If the particles are produced by the interaction of a particle with the material of the detector, the vertex is a secondary interaction vertex. An example is bremsstrahlung, the radiation of a photon by an electron in the electric field of a nucleus.

The primary vertex in an event is usually easy to find, especially if prior information about its location is available, such as the beam profile or the target position. A notable exception is the invisible primary vertex of a $\Upsilon(4S) \to B^0\bar{B}^0$ decay at a B factory. On the other hand, secondary decay vertices of short-lived particles may be hard to find, as some of the decay products may also be compatible with the primary vertex. An example is shown in Fig. 7. In this example one of the primary tracks passes very close to the secondary vertex (top panel). This could lead to a wrong assignment of this track. The bottom panel shows that some of the secondary tracks, when extrapolated back into beam tube, pass very close to the primary vertex. This could result in a wrong assignment of these tracks. In order to achieve optimal separation of primary and secondary vertices, the spatial resolution of the innermost layers and the minimization of the material are of the utmost importance.

1. Cluster finding

Clustering methods are based on a distance matrix or a similarity matrix of the objects to be classified. A cluster is then a group with small distances (large similarities) inside the group and large distances (small similarities) to objects outside the group. The distance measure reflects only the geometry of the tracks. Hierarchical clustering builds a tree of clusterings. Agglomerative methods start at the leaves of the tree, i.e., the single objects, while divisive methods start at the root, the totality of all objects.

In the simplest case, the clustering can be reduced to a one-dimensional problem. An example is primary vertex finding in the pixel (PIX) detector of the CMS experiment at the LHC (Cucciarelli, 2005). The input of the algorithm consists of triplets of pixel hits that are compatible with a track hypothesis. From each triplet the longitudinal impact point z_{IP} and its error are computed. Clusters are found by two methods. With the histogramming method, the impact points are filled into a histogram. Empty bins are discarded, and the nonempty bins are scanned along z. A cluster is defined as a contiguous

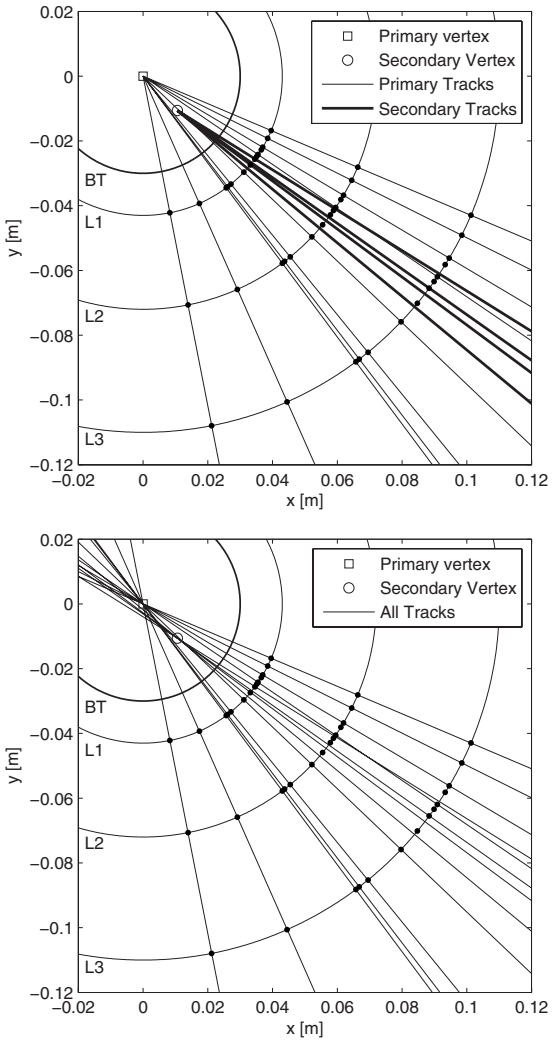

FIG. 7. A hard vertex finding problem. Top: The primary vertex with ten tracks and a secondary vertex with four tracks at a distance of 1 cm from the primary vertex. Bottom: All tracks extrapolated into the beam tube. BT, beam tube; L1, layer 1, etc.

set of consecutive bins separated by a distance smaller than a threshold Δz. After a cleaning procedure, the z positions of the clusters are recomputed as a weighted z_{IP} average of the remaining tracks, the weights being the inverse squared errors of the longitudinal impact points. The primary vertex is identified as the cluster with the largest sum of squared transverse momenta.

The second method described by Cucciarelli (2005) is a hierarchical clustering of the divisive kind. The tracks are ordered by increasing z_{IP}, and the ordered list is scanned. A cluster is terminated when the gap between two consecutive tracks exceeds a threshold, and a new cluster is started. For each initial cluster, an iterative procedure is applied to discard incompatible tracks. The discarded tracks are recovered to form a new cluster, and the same procedure is applied again until there are less than two remaining tracks.

In the general case, clustering proceeds in space. Various clustering methods of both the hierarchical and nonhierarchical types have been evaluated in the context of vertex finding (Waltenberger, 2004). In hierarchical agglomerative clustering each track starts out as a single cluster. Clusters are merged iteratively on the basis of a distance measure. The shortest distance in space between two tracks is peculiar insofar as it does not satisfy the triangle inequality: if tracks a and b are close and tracks b and c are close, it does not follow that tracks a and c are close as well. The distance between two clusters of tracks should therefore be defined as the maximum of the individual pairwise distances, known as complete linkage in the clustering literature. Divisive clustering starts out with a single cluster containing all tracks. Further division of this cluster can be based on repeated vertex estimation with identification of outliers. Some examples of the approach are described in Sec. II.D.3.

2. Topological vertex finding

A very general topological vertex finder (ZVTOP) was proposed by Jackson (1997). It is related to the Radon transform, which is a continuous version of the Hough transform used for track finding (see Sec. II.A). The search for vertices is based on a function $V(\boldsymbol{v})$ which quantifies the probability of a vertex at location $\boldsymbol{v}$. For each track a Gaussian probability tube $f_i(\boldsymbol{v})$ is constructed by

$$f_i(\boldsymbol{v}) = \exp\left[-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{r})^{\mathrm{T}}\boldsymbol{V}_i^{-1}(\boldsymbol{v}-\boldsymbol{r})\right], \tag{26}$$

where $\boldsymbol{r}$ is the point of closest approach of track i to point $\boldsymbol{v}$ and $\boldsymbol{V}_i$ is the covariance matrix of the track at $\boldsymbol{r}$.

The vertex function $V(\boldsymbol{v})$ is defined taking into account that in the neighborhood of a vertex the value of $f_i(\boldsymbol{v})$ must be significant for at least two tracks,

$$V(\boldsymbol{v}) = \sum_{i=0}^{n} f_i(\boldsymbol{v}) - \sum_{i=0}^{n} f_i^2(\boldsymbol{v}) \Bigg/ \sum_{i=0}^{n} f_i(\boldsymbol{v}). \tag{27}$$

Due to the second term on the right-hand side, $V(\boldsymbol{v}) \approx 0$ in regions where $f_i(\boldsymbol{v})$ is significant for only one track. The form of $V(\boldsymbol{v})$ can be modified to fold in known physics information about probable vertex locations. For instance, $V(\boldsymbol{v})$ can be augmented by a further function $f_0(\boldsymbol{v})$ describing the location and spread of the interaction point. In addition, $V(\boldsymbol{v})$ may be modified by a factor dependent on the angular location of the point $\boldsymbol{v}$.

Vertex finding amounts to finding local maxima of the function $V(\boldsymbol{v})$. The search starts at the calculated maxima of the products $f_i(\boldsymbol{v})f_j(\boldsymbol{v})$ for all track pairs. For each of these points the nearest maximum of $V(\boldsymbol{v})$ is found. These maxima are clustered together to form candidate vertex regions.

3. Minimum spanning tree

A recent extension to the ZVTOP algorithm uses the graph-theoretical concept of the minimum spanning tree (MST) (Kruskal, 1956). The ZVMST vertex finder (Hillert, 2008) has two stages. In the first, a small number of likely vertex positions are chosen on the basis of function (27). In the second, tracks are assigned to these preliminary vertices.

The first stage of ZVMST starts by forming all possible two-track combinations and discarding bad ones by means of a χ^2 cut. The retained combinations are used to set up a weighted graph, where each node represents a track, each edge represents a successful vertex fit, and the weight is equal to the inverse of the vertex function at this vertex. The graph is passed to an MST algorithm, which prunes the graph to a tree such that the total sum of the weights is minimized. The vertices corresponding to surviving edges are then merged to candidate vertices on the basis of their proximity in space. In the second stage of ZVMST tracks are associated to the candidate vertices based on both the values of the probability tubes [see Eq. (26)] and the values of the vertex functions [see Eq. (27)] at the candidate positions.

4. Feed-forward neural networks

Feed-forward neural networks, also called multilayer perceptrons, are classifiers that learn their decision rules on a training set of data with known classification. If the data at hand do not conform to the properties of the training sample, the network cannot cope with this situation. Such networks therefore cannot be considered as adaptive.

Primary vertex finding with a feed-forward neural network was proposed by Lindsey and Denby (1991). The input to the network was provided by the drift times of tracks in a planar drift chamber parallel to the colliding proton-antiproton beams. The chamber was divided into overlapping 18-wire subsections. The 18 wires correspond to the 18 input units of the network. The hidden layer of the network had 128 neurons and the output

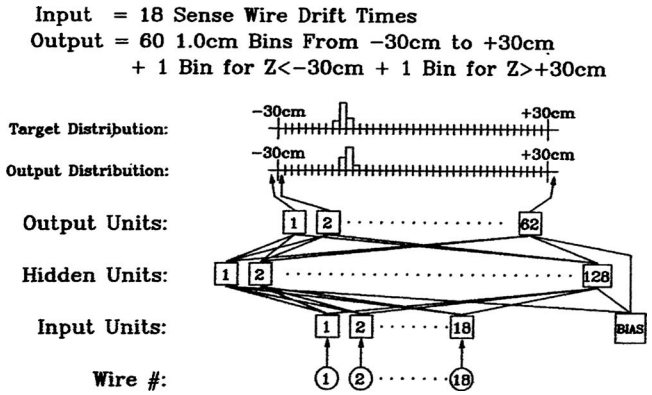

FIG. 8. The neural network architecture used to determine the vertex of tracks in 18-wire z-chamber subsections. All input units and the bias unit are connected to all hidden units. All hidden units and the bias unit are connected to all output units. Only a few of the connections are shown. The bias unit has an output activation fixed to 1.0. From Lindsey and Denby, 1991.

layer had 62 units, corresponding to 62 1-cm bins along the beam line (see Fig. 8). The network was trained by back-propagation on 12 000 patterns, a mixture of events with single tracks, single tracks with noise hits, double tracks, and empty events. The performance was reported to be nearly as good as the conventional offline fitting algorithm.

A more advanced network was proposed for vertex finding in the ZEUS central tracking detector (Dror and Etzion, 2000). The network is based on stepwise changes in the representation of the data. It moves from the input points to local line segments, then to local arcs, and finally to global arcs. The z coordinate of the vertex is computed by a second three-layer network running in parallel. The resolution of the vertex coordinate z was reported to be better by almost a factor of 2 compared to the conventional histogramming method.

D. Vertex fitting

1. Least-squares methods for vertex fitting

Classical methods of vertex fitting are similar in many respects to track fitting methods. The parameters to be estimated are the common vertex $\boldsymbol{v}$ of a set of n tracks and the momentum vectors $\boldsymbol{p}_i$ of the participating tracks. The "measurements" are the estimated track parameters $\tilde{\boldsymbol{q}}_i$ along with their covariance matrices $\boldsymbol{V}_i=\boldsymbol{G}_i^{-1}$, $i=1,\ldots,n$, at some reference surface (Fig. 9).

The model for a global least-squares estimator can be written down in the following form:

$$\begin{pmatrix}\tilde{\boldsymbol{q}}_1\\ \vdots\\ \tilde{\boldsymbol{q}}_n\end{pmatrix}=\begin{pmatrix}\boldsymbol{h}_1(\boldsymbol{v},\boldsymbol{p}_1)\\ \vdots\\ \boldsymbol{h}_n(\boldsymbol{v},\boldsymbol{p}_n)\end{pmatrix}+\begin{pmatrix}\boldsymbol{\epsilon}_1\\ \vdots\\ \boldsymbol{\epsilon}_n\end{pmatrix}, \tag{28}$$

where $\boldsymbol{q}_i=\boldsymbol{h}_i(\boldsymbol{v},\boldsymbol{p}_i)$ is the deterministic track model of track i and $\boldsymbol{\epsilon}_i$ is the estimation error of track i. If there is multiple scattering between the positions of the vertex and of the track parameters, its effect has to be included

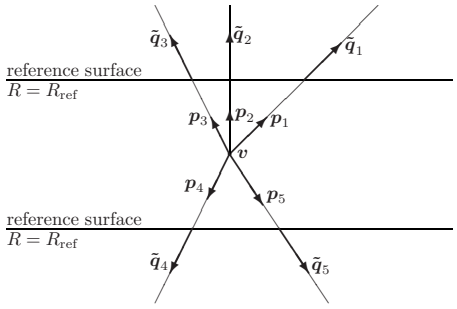

FIG. 9. Vertex parameters $\boldsymbol{v}$ (position), $\boldsymbol{p}_1,\ldots,\boldsymbol{p}_5$ (momentum vectors), and estimated track parameters $\tilde{\boldsymbol{q}}_1,\ldots,\tilde{\boldsymbol{q}}_5$ for a vertex with five tracks. The reference surface is a cylinder with radius $R=R_{\mathrm{ref}}$.

in $\boldsymbol{\epsilon}_i$ as well. As the track estimates are stochastically independent, the joint covariance matrix of all $\boldsymbol{\epsilon}_i$ is block diagonal.

Least-squares estimation of this model requires minimization of the following objective function:

$$F(\boldsymbol{v},\boldsymbol{p}_1,\ldots,\boldsymbol{p}_n)=\sum_{i=1}^{n}\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{G}_i\boldsymbol{e}_i,\quad \boldsymbol{e}_i=\tilde{\boldsymbol{q}}_i-\boldsymbol{q}_i. \tag{29}$$

Minimization of the objective function can proceed in several ways. A detailed exposition of nonlinear least-squares estimation can be found, for example, in Bates and Watts (1988). The most popular estimation method is the Gauss-Newton method. Starting from approximate parameter values $\check{\boldsymbol{v}}$ and $\check{\boldsymbol{p}}_i$ for all i, the track model can be approximated by an affine function

$$\boldsymbol{q}_i\approx\boldsymbol{c}_i+\boldsymbol{A}_i\boldsymbol{v}+\boldsymbol{B}_i\boldsymbol{p}_i, \tag{30}$$

with

$$\boldsymbol{A}_i=\left.\frac{\partial\boldsymbol{h}_i(\boldsymbol{v},\boldsymbol{p}_i)}{\partial\boldsymbol{v}}\right|_{\check{\boldsymbol{v}},\check{\boldsymbol{p}}_i},\quad \boldsymbol{B}_i=\left.\frac{\partial\boldsymbol{h}_i(\boldsymbol{v},\boldsymbol{p}_i)}{\partial\boldsymbol{p}_i}\right|_{\check{\boldsymbol{v}},\check{\boldsymbol{p}}_i},$$
$$\boldsymbol{c}_i=\boldsymbol{h}_i(\check{\boldsymbol{v}},\check{\boldsymbol{p}}_i)-\boldsymbol{A}_i\check{\boldsymbol{v}}-\boldsymbol{B}_i\check{\boldsymbol{p}}_i. \tag{31}$$

Using this approximation, Eq. (28) can be rewritten as a linear model,

$$\begin{pmatrix}\tilde{\boldsymbol{q}}_1-\boldsymbol{c}_1\\ \tilde{\boldsymbol{q}}_2-\boldsymbol{c}_2\\ \vdots\\ \tilde{\boldsymbol{q}}_n-c_n\end{pmatrix}=\underbrace{\begin{pmatrix}\boldsymbol{A}_1 & \boldsymbol{B}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{A}_2 & \boldsymbol{0} & \boldsymbol{B}_2 & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{A}_n & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{B}_n\end{pmatrix}}_{\boldsymbol{M}}\begin{pmatrix}\boldsymbol{v}\\ \boldsymbol{p}_1\\ \vdots\\ \boldsymbol{p}_n\end{pmatrix}+\begin{pmatrix}\boldsymbol{\epsilon}_1\\ \vdots\\ \boldsymbol{\epsilon}_n\end{pmatrix}. \tag{32}$$

Because of the special structure of the model matrix $\boldsymbol{M}$ the minimization of objective function (29) results in a closed-form solution (Billoir *et al.*, 1985),

$$\tilde{\boldsymbol{v}}_n=\boldsymbol{C}_n\sum_{i=1}^{n}\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{G}_i^{\mathrm{B}}(\tilde{\boldsymbol{q}}_i-\boldsymbol{c}_i), \tag{33}$$

$$\mathrm{var}(\tilde{\boldsymbol{v}}_n) = \boldsymbol{C}_n = \left(\sum_{i=1}^{n} \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i \right)^{-1},$$

with

$$\begin{aligned} \boldsymbol{G}_i^{\mathrm{B}} &= \boldsymbol{G}_i - \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i, \\ \boldsymbol{W}_i &= (\boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{B}_i)^{-1}. \end{aligned} \tag{34}$$

The subscript n indicates that the vertex estimate $\tilde{\boldsymbol{v}}_n$ is based on all of the n tracks.

In general, the procedure has to be iterated. The model equation [Eq. (28)] is expanded at the new estimate, and the estimate is recomputed until convergence is obtained. The formulas required for the implementation of two important cases, fixed-target configuration and solenoidal configuration, are given by Harr (1995).

Once $\tilde{\boldsymbol{v}}_n$ is known, the estimated track momenta and the full covariance matrix can be computed,

$$\begin{aligned} \tilde{\boldsymbol{p}}_i^n &= \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \tilde{\boldsymbol{v}}_n), \\ \mathrm{var}(\tilde{\boldsymbol{p}}_i^n) = \boldsymbol{D}_i^n &= \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_n \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i, \\ \mathrm{cov}(\tilde{\boldsymbol{p}}_i^n, \tilde{\boldsymbol{v}}_n) = \boldsymbol{E}_i^n &= -\boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_n, \quad i = 1, \ldots, n. \end{aligned} \tag{35}$$

The superscript n indicates that the estimated track momentum of track i uses information from all n tracks via the vertex estimate $\tilde{\boldsymbol{v}}_n$.

The estimates can also be computed progressively, resulting in an extended Kalman filter (Billoir *et al.*, 1985; Frühwirth, 1987; Catlin, 1989). Assume that the vertex $\tilde{\boldsymbol{v}}_{i-1}$ has been estimated using tracks 1 to $i-1$. Track i is added via the following update equations:

$$\begin{aligned} \tilde{\boldsymbol{v}}_i &= \boldsymbol{C}_i [\boldsymbol{C}_{i-1}^{-1} \tilde{\boldsymbol{v}}_{i-1} + \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i^{\mathrm{B}} (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i)], \\ \tilde{\boldsymbol{p}}_i &= \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \tilde{\boldsymbol{v}}_i), \\ \boldsymbol{C}_i &= (\boldsymbol{C}_{i-1}^{-1} + \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{A}_i)^{-1}, \\ \boldsymbol{D}_i &= \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_i \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i, \\ \boldsymbol{E}_i &= -\boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_i. \end{aligned} \tag{36}$$

Apart from small numerical effects, the final result does not depend on the order in which the tracks are added. The latter is therefore arbitrary.

The smoother associated to this filter is tantamount to recomputing the track momenta using the last vertex estimate $\tilde{\boldsymbol{v}}_n$, i.e., Eqs. (35). For implementations of the full Kalman-filter formalism in the LHC experiments ATLAS and CMS, see Waltenberger (2004) and Wildauer (2006).

Using the estimated vertex position and track momenta, the estimated track parameters can be updated to

$$\tilde{\boldsymbol{q}}_i^n = \boldsymbol{h}(\tilde{\boldsymbol{v}}_n, \tilde{\boldsymbol{p}}_i^n), \tag{37}$$

$$\mathrm{var}(\tilde{\boldsymbol{q}}_i^n) = \boldsymbol{B}_i \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}} + \boldsymbol{V}_i^{\mathrm{B}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_n \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{V}_i^{\mathrm{B}},$$

with

$$\boldsymbol{V}_i^{\mathrm{B}} = \boldsymbol{V}_i - \boldsymbol{B}_i \boldsymbol{W}_i \boldsymbol{B}_i^{\mathrm{T}}. \tag{38}$$

The updated track parameters are no longer independent.

The χ^2 statistic of the final estimate is given by

$$\chi^2 = \sum_{i=1}^{n} \boldsymbol{r}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{r}_i, \tag{39}$$

with the residuals $\boldsymbol{r}_i = \tilde{\boldsymbol{q}}_i - \tilde{\boldsymbol{q}}_i^n$. It can be used to test the vertex hypothesis, i.e., the correctness of model (28). If the test fails, however, there is no information about which track(s) may have caused the failure.

The estimated track parameters $\tilde{\boldsymbol{q}}_i$ are frequently given at the innermost detector surface or at the beam tube. If the $\tilde{\boldsymbol{q}}_i$ are propagated to the vicinity of the presumed vertex, the vertex estimation can be speeded up by applying some approximations, especially if the magnetic field is sufficiently homogeneous.

The "perigee" parametrization for helical tracks was introduced by Billoir and Qian (1992) with a correction by Billoir and Qian (1994). The track is parametrized around the point of closest approach (the perigee point $\boldsymbol{v}^{\mathrm{P}}$) of the helix to the z axis. The variation of the transverse errors along the track is neglected in the vicinity of the perigee, and track direction and curvature at the vertex are assumed to be constant. The approximate objective function of the vertex fit can then be written entirely in terms of the perigee points,

$$F(\boldsymbol{v}) = \sum_{i=1}^{n} (\boldsymbol{v}_i^{\mathrm{P}} - \boldsymbol{v})^{\mathrm{T}} \boldsymbol{T}_i (\boldsymbol{v}_i^{\mathrm{P}} - \boldsymbol{v}), \tag{40}$$

where $\boldsymbol{T}_i$ is a weight matrix of rank 2. The vertex estimate is then

$$\tilde{\boldsymbol{v}} = \left(\sum_{i=1}^{n} \boldsymbol{T}_i \right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{T}_i \boldsymbol{v}_i^{\mathrm{P}} \right). \tag{41}$$

The Jacobians required to compute the $\boldsymbol{T}_i$ are given by Billoir and Qian (1992, 1994). For an implementation in ATLAS, see Wildauer (2006) and Piacquadio *et al.* (2008).

A further simplification was proposed by Karimäki (1997). The track is approximated by a straight line in the vicinity of the vertex. The estimated track parameters are transformed to a coordinate system the x axis of which is parallel to the track. The vertex is then estimated by minimizing the sum of the weighted transverse distances of the tracks to the vertex. The resulting objective function has the same form as in Eq. (40), again with weight matrices of rank 2. The estimate is exact for straight tracks. The method has been implemented in CMS (Waltenberger, 2004).

2. Robust vertex fitting

In experimental reality vertex candidates as produced by the vertex finder are often contaminated by outliers. Outliers can be broadly classified into two categories. In the first category are mismeasured tracks, i.e., tracks for which the deviation of the estimated track parameters from the true track parameters is larger than indicated by the covariance matrix of the track. There may be several reasons for that, including extraneous observations resulting from a mistake in the track finding, a wrong estimate of the material the track has crossed, or a wrong estimate of the covariance matrix of an observation. In the second category are extraneous tracks not belonging to the vertex at all, e.g., a primary track attached by mistake to a secondary vertex, or a background track attached to either a primary or a secondary vertex.

As in the case of track reconstruction, there are two different ways of dealing with outliers. Either outliers are identified and removed or the influence of outliers is reduced by using robust estimators. Outliers can be identified by inspecting their χ^2 increment with respect to the vertex estimated from the other tracks,

$$\chi_i^2 = \boldsymbol{r}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{r}_i + (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \tilde{\boldsymbol{v}}_n)^{\mathrm{T}} \cdot \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i (\boldsymbol{C}_n^i)^{-1} \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i^{\mathrm{B}} (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \tilde{\boldsymbol{v}}_n), \tag{42}$$

with $\boldsymbol{C}_n^i = \boldsymbol{C}_n - \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i$ the covariance matrix of the vertex fitted from all tracks except track i. If there are no outliers, χ_i^2 is χ^2 distributed with $m = \dim(\tilde{\boldsymbol{q}}) - \dim(\tilde{\boldsymbol{v}})$ degrees of freedom. As outliers should give significantly larger values of χ_i^2, the latter can be used to test for the presence of outliers. If there are several outliers, the fitted vertex is always biased even if one of the outliers is removed. In this case it is better to reduce the influence of outliers already in the estimate itself by constructing a robust estimator.

One of the earliest proposals for a robust vertex fit (Frühwirth *et al.*, 1996) is an M-estimator with Huber's ψ function (Huber, 1981). It can be implemented as an iterated reweighted least-squares estimator. The initial vertex estimate is a plain least-squares estimate. Then, for each track, the residuals are rotated to the eigensystem of the covariance matrix of the track, and weight factors are computed according to

$$w_i = \begin{cases} 1, & |r_i| \leqslant c\sigma_i \\ c\sigma_i / |r_i|, & |r_i| > c\sigma_i, \end{cases} \tag{43}$$

where r_i is one of the residuals in the rotated frame, σ_i is the standard deviation in the rotated frame, and c is the robustness constant, usually chosen between 1 and 3. The weight factors are applied and the estimate is recomputed. The entire procedure is iterated until convergence.

A similar kind of reweighted least-squares estimator was proposed by Agakichiev *et al.* (1997). The weights are computed according to Tukey's bisquare function (Hampel *et al.*, 1986),

$$w_i = \begin{cases} \left(1 - \dfrac{r_i^2/\sigma_i^2}{c^2}\right)^2 & \text{for } |r_i| \leqslant c\sigma_i \\ 0 & \text{otherwise}, \end{cases} \tag{44}$$

where r_i^2 is the squared residual of track i with respect to the vertex, σ_i^2 is its variance, and c is again the robustness constant. The estimator is now equivalent to a redescending M-estimator (Hampel *et al.*, 1986) and therefore less sensitive to outliers than Huber's M-estimator. The same weights were proposed by Golutvin *et al.* (2000) for robust track fitting.

Another obvious candidate for robust vertex estimation is the least trimmed squares (LTS) estimator (Rousseeuw and Leroy, 1987; D'Hondt *et al.*, 2004; Chabanat *et al.*, 2005). With the LTS estimator, the objective function of Eq. (29) is replaced by

$$F(\boldsymbol{v}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) = \sum_{i=1}^{\lfloor hn \rfloor} \boldsymbol{e}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{e}_i, \quad \boldsymbol{e}_i = \tilde{\boldsymbol{q}}_i - \boldsymbol{q}_i, \tag{45}$$

where h is the user-defined fraction of tracks to be included in the estimate. Finding the subset of tracks that minimizes F is a combinatorial problem, which makes the LTS estimator much more difficult (and slower) to compute than the least-squares estimator. In addition, the fraction h is fixed and has to be specified in advance. As a consequence, tracks are thrown away even if there are no outliers, and the precision of the estimate suffers. Both problems can be overcome by the introduction of the adaptive vertex estimator (see Sec. III.F).

3. Vertex finding by iterated fitting

The test on outliers based on the χ^2 increment χ_i^2 [see Eq. (42)] can be expanded to a full-blown vertex finder by recursively identifying and removing outliers (Speer *et al.*, 2006c). The algorithm has been called "trimmed Kalman filter" but should not be confused with the LTS estimator (see Sec. II.D.2).

First, all input tracks are fitted to a common vertex using a standard least-squares estimator (see Sec. II.D.1). The χ^2 increment is computed for all tracks, the track with the largest increment is removed from the vertex, and the vertex is refitted. This procedure is repeated until the χ^2 increment of all tracks is below a given threshold. Once a track has been rejected, it is not included again in the vertex.

When a vertex has been established, the entire procedure is repeated on the set of the remaining tracks. The iteration stops when no vertex with at least two tracks can be successfully fitted.

If there are no outliers and if all tracks are correctly estimated at the track fitting stage, the χ^2 increment is distributed according to a χ^2 distribution with $m = \dim(\tilde{\boldsymbol{q}}) - \dim(\tilde{\boldsymbol{v}})$ degrees of freedom. If the threshold mentioned above is set to the $1-\alpha$ quantile of the χ^2 distribution with m degrees of freedom, then under the

null hypothesis of no outliers the probability of rejecting a randomly chosen track is equal to α and the probability of rejecting the track with the largest χ^2 increment is approximately equal to $1-(1-\alpha)^n$ if the number of tracks n is large enough so that the correlations between the χ^2 increments can be neglected.

If there are outliers, the χ^2 increments are no longer χ^2 distributed, and the probability of rejecting a randomly chosen good track may be well above α. The respective probabilities of falsely rejecting good tracks and correctly identifying outliers can no longer be calculated analytically and have to be determined by simulation studies.

The iterative identification of outliers is therefore a lengthy and somewhat unreliable procedure, especially if there are several outliers, resulting in a severe bias on the estimated vertex position. These problems can be overcome by employing not a least-squares estimator but an adaptive estimator in each stage of the iteration (see Sec. III.F).

III. ADAPTIVE METHODS

The first attempt to equip track reconstruction methods with adaptive behavior was the application of the Hopfield network to track finding (Denby, 1988; Peterson, 1989). As it turned out to be difficult to embed the correct physical track model into the Hopfield network, the next step was the elastic arms or deformable template algorithm (Ohlsson *et al.*, 1992) and a related method called elastic tracking (Gyulassy and Harlander, 1991). At this point the traditional boundaries between pattern recognition (track finding) and parameter estimation (track fitting) started to dissolve. Both methods, however, required numerical minimization of a complicated energy function. The realization that an alternative way of minimization was provided by the EM algorithm (Dempster *et al.*, 1977) then paved the way to adaptive least-squares estimators in general and to an adaptive version of the Kalman filter in particular (Frühwirth and Strandlie, 1999). Even the concept of annealing, native to the world of neural networks and combinatorial optimization, could be transplanted in a natural way into the world of stochastic filters. As vertex reconstruction is similar to track reconstruction in many respects, adaptive methods developed for track finding and fitting could be applied to vertex finding and fitting almost one to one.

Adaptive methods for track and vertex reconstruction tend to have a certain set of features in common. The most salient ones are the following.

- After being initialized at some position, the track or vertex moves during an iterative procedure due to some defined dynamical scheme. The dynamical scheme often uses the information from the data several times during the iterations. This can also be regarded as a sequential exploration of several hypotheses about the correct values of the set of parameters describing the track or vertex.
- The observations—detector hits in the case of track fitting, reconstructed tracks in the case of vertex fitting—can also have a weight attached to them, potentially depending on the positions of other observations, thereby opening up the possibility of soft assignment of the observations to the track or vertex. In the case of track fitting, sets of mutually exclusive observations, e.g., several hits in the same detector layer, can compete for inclusion in the track in the sense that hits close to the track tend to downweight the influence of hits that are further away in the same detector layer.
- The dynamics of the track or vertex can often be regarded as an iterative strategy for locating the minimum of an energy hypersurface in the space of the track or vertex parameters. This energy surface often has several local minima. In order to increase the probability of reaching the global minimum, some of the methods introduce the concept of temperature and annealing. Typically, an algorithm is initiated at a high temperature, and the temperature parameter is decreased ("cooled") during the iterations. The effect is that the energy surface is smoothed out at high temperatures, whereas the original structure of the energy landscape shows up in the late stages of the iterations when the temperature is low.
- A nonlinear filter can explore a set of track or vertex hypotheses in parallel rather than in sequence. The final result is calculated by collapsing the surviving hypotheses into a single Gaussian distribution. The concepts of weights, soft assignment, and competition are therefore also relevant for nonlinear filters.

A. Hopfield neural networks

Artificial neural networks (Hertz *et al.*,1991) constitute a computational paradigm which is by now well established. Such networks are used in a range of different application areas, for instance, within pattern classification and combinatorial optimization problems. The first application in high-energy physics dates back to 1989. Earlier reviews of the applications of neural networks in high-energy physics can be found in Kisel *et al.* (1993) and Denby (1999).

For the solution of combinatorial optimization problems Hopfield networks (Hopfield, 1982) have emerged as particularly powerful tools. The standard dynamics of a Hopfield network

$$S_i = \mathrm{sgn}\left(\sum_j T_{ij} S_j\right), \tag{46}$$

where T_{ij} is the connection weight between neuron i and neuron j and the sign function $\mathrm{sgn}(x)$ is taken to be

$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geqslant 0 \\ -1, & x < 0, \end{cases} \tag{47}$$

leads to a local minimum of the energy function

$$E = -\frac{1}{2}\sum_i \sum_j T_{ij} S_i S_j \tag{48}$$

with respect to the configuration of the set of binary-valued neurons $\{S_i\}$. The general solution to the problem is tantamount to finding the global minimum of the energy function.

A benchmark problem in combinatorial optimization is the traveling salesman problem, which asks for the shortest path through N cities positioned at a set of known coordinates. From Hopfield and Tank (1985) the traveling salesman problem was formulated as a minimization problem of an energy function of a Hopfield network. Since the local minimum found by the standard network dynamics can provide solutions quite far from the desired global optimum, it was suggested to use a smooth update function, the hyperbolic tangent, inspired by mean-field theory from statistical mechanics. The mean-field theory update equations were rigorously derived by means of a saddle-point approximation by Peterson and Anderson (1987). The main idea behind this approximation is that the behavior of the partition function Z,

$$Z = \sum_{\boldsymbol{S}} e^{-E(\boldsymbol{S})/T}, \tag{49}$$

where $\boldsymbol{S}=\{S_1, S_2, \ldots, S_N\}$ is a configuration of the N neurons in the network is dominated by the low-energy configurations of $E(\boldsymbol{S})$. These configurations are searched for by first replacing the sum over different configurations $S=\pm 1$ by integrals over continuous variables U and V, giving

$$Z = c\prod_i \int_{-\infty}^{\infty} dV_i \int_{-i\infty}^{i\infty} dU_i e^{-E'(\boldsymbol{V},\boldsymbol{U},T)}, \tag{50}$$

where c is a normalization constant and the effective energy E' is given by

$$E'(\boldsymbol{V},\boldsymbol{U},T) = \frac{E(\boldsymbol{V})}{T} + \sum_i [U_i V_i - \ln(\cosh\, U_i)]. \tag{51}$$

The saddle points of Z are found by requiring that the partial derivatives of E' with respect to both mean-field variables U_i and V_i are zero, giving the update equations

$$V_i = \tanh\left(\sum_j T_{ij} V_j / T\right). \tag{52}$$

The energy landscape of E' is smoother than that of the original energy E particularly at high temperatures T. The strategy of finding low-energy configurations is thus to initiate the search at a high temperature and iterate Eq. (52) until convergence. The temperature is lowered, and a new minimum configuration of the mean-field variables is found, starting out from the configuration obtained by the previous iteration. The whole procedure is repeated at successively lower temperatures, approaching the zero-temperature limit in the end. In this limit, the mean-field equations reduce to the standard dynamics of the Hopfield network. Since the mean-field equations are solved at a sequence of decreasing temperatures, the procedure is called mean-field annealing.

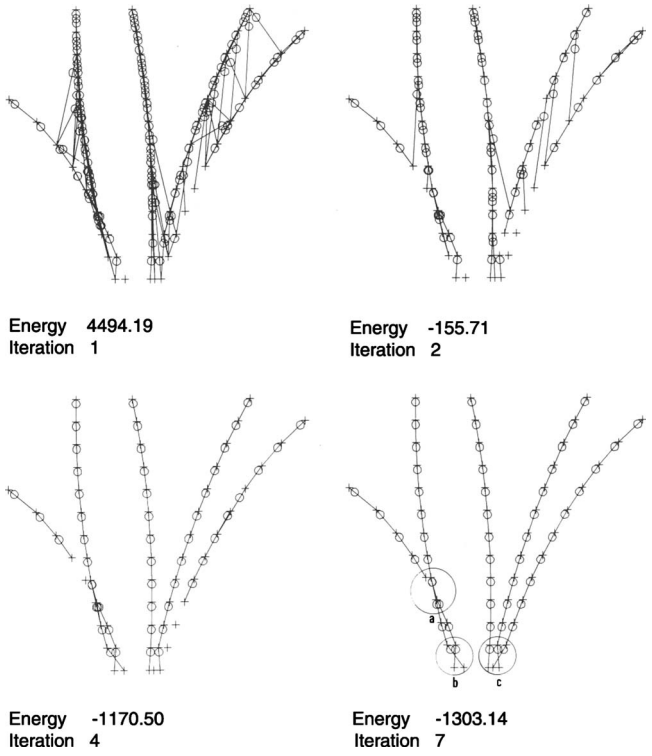

FIG. 10. Track finding with a model neural network. Total network energy and iteration number are given. Measured space points are represented by crosses and neurons by segments joining points with a circle at the neuron head indicating direction. Only neurons with output values greater than 0.1 are drawn. In practice, most neurons are found to have values near either 0 or 1. At convergence (final frame, bottom right), the reconstruction is not perfect. An example of missing or illegal neurons can be found in the circle marked a; incorrect choices of neurons can be observed in the circles marked b and c. From Denby, 1988.

It was realized independently by Denby (1988) and Peterson (1989) that the problem of finding track candidates in a high-energy physics particle detector could be formulated as the problem of minimizing an energy function of the Hopfield type. In this model, called the Denby-Peterson network, the neurons correspond to links between measurements in adjacent detector layers, and the weights quantify the probability of two adjoining links belonging to the same track. If two such adjoining links point in approximately the same direction, the energy is decreased and if two links start from or end up in the same measurement, the energy is increased. An overall constraint, limiting the total number of active neurons to the overall number of measurements, is also included in the energy function. The network is run to convergence with mean-field annealing. Figure 10 shows an example of a Denby-Peterson network in various stages of its evolution. At the end some shortcomings, such as missing or illegal neurons and incorrect choices, can be observed. Of course the Hopfield dynamics is just one way of minimizing the energy function. Alternatives

to mean-field annealing have been explored but found to be inferior (Diehl *et al.*, 1997).

The first experimental implementation of a Hopfield network was done in the ALEPH experiment (Stimpfl-Abele and Garrido, 1991) and found to give results compatible with those produced by the standard track finder used in the experiment. A simplified version of a Hopfield network has also been tested on real data from the ARES spectrometer (Baginyan *et al.*, 1994). More recently, the method has been tried on simulated data from the ALICE experiment (Badalà *et al.*, 2003, 2004). Used in combination with the standard Kalman-filter based track finding procedure, it was shown to increase the track finding efficiency as compared to a stand-alone application of the Kalman filter.

B. Elastic nets and deformable templates

An alternative solution to the traveling salesman problem is the application of the elastic net method (Durbin and Willshaw, 1987). In this method, the initial path is a smooth curve, which in general does not pass through any of the cities. By an iterative procedure, the path is gradually deformed through the influence of two different forces, whose relative strengths are governed by two constants α and β. If the coordinates of city i are given by $\boldsymbol{x}_i$ and the coordinates of a typical point j along the path are given by $\boldsymbol{y}_j$, the change $\Delta\boldsymbol{y}_j$ in an iteration is

$$\Delta\boldsymbol{y}_j = \alpha\sum_i w_{ij}(\boldsymbol{x}_i - \boldsymbol{y}_j) + \beta K(\boldsymbol{y}_{j+1} - 2\boldsymbol{y}_j + \boldsymbol{y}_{j-1}), \tag{53}$$

where K is a constant. One of the forces attracts the path toward the cities, whereas the other one tries to minimize the total length of the path. The coefficient w_{ij} is the strength of the connection between city i and point j. It is normalized,

$$w_{ij} = \varphi(|\boldsymbol{x}_i - \boldsymbol{y}_j|, K) \Big/ \sum_k \varphi(|\boldsymbol{x}_i - \boldsymbol{y}_k|, K). \tag{54}$$

If $\varphi(d,K)$ is taken to be Gaussian,

$$\varphi(d,K) = e^{-d^2/2K^2}, \tag{55}$$

an energy function E can be defined as

$$E = -\alpha K\sum_i \ln\left(\sum_j \varphi(|\boldsymbol{x}_i - \boldsymbol{y}_j|, K)\right) + \beta\sum_j |\boldsymbol{y}_{j+1} - \boldsymbol{y}_j|^2 \tag{56}$$

such that the update of the position $\boldsymbol{y}_j$ can be regarded as a gradient descent of the energy function in the coordinate space of $\boldsymbol{y}_j$,

$$\Delta\boldsymbol{y}_j = -K\frac{\partial E}{\partial \boldsymbol{y}_j}. \tag{57}$$

The value of K is lowered during the iterations, causing the path to eventually pass through all of the cities. On problems with a quite large number of cities, the elastic net method is superior to the Hopfield network approach (Durbin and Willshaw, 1987).

The elastic net method has been successfully applied to track fitting in a drift-tube detector in the NEMO experiment (Kisel *et al.*, 1997). For this application, three different forces are acting during the iterations. One draws the track to the edge of the drift circle defining the possible positions of the measurement in the tube, the other smoothes out the track, and the third attracts the two lines constituting the envelope of the track toward each other. An example is shown in Fig. 11.

An application of the elastic net to vertex finding is described by Kisel, Konotopskaya, and Kovalenko (1997). As the vertex can be anywhere in a large target (diameter $\approx$10 cm) there is no good initial approximation to the vertex position. The vertex is therefore defined as the center of the area with maximum track density. The vertex finder is based on an elastic ring and uses two types of forces: attraction of the ring to all tracks, shifting toward the area with high track density, and attraction to the nearest tracks, localizing the vertex region. More recently, the elastic net method has also been applied to ring finding in a ring imaging Čerenkov detector in the CBM experiment (Gorbunov and Kisel, 2005, 2006).

An important generalization of the elastic net method has been developed by Yuille (1990), formulating the traveling salesman problem as an assignment problem, with an energy function containing binary assignment variables V_{ij}. These variables are equal to 1 if city i is assigned to point j and 0 otherwise. In order to assign a city to only one point, the constraint $\Sigma_j V_{ij}=1$ is introduced. The energy function E reads

$$E[\{V_{ij}\},\{\boldsymbol{y}_j\}] = \sum_{ij} V_{ij}|\boldsymbol{x}_i - \boldsymbol{y}_j|^2 + \nu\sum_j |\boldsymbol{y}_j - \boldsymbol{y}_{j+1}|^2, \tag{58}$$

where ν is a positive constant. Finding a minimum of E with respect to all possible and allowed values of $\{V_{ij}\}$ and $\{\boldsymbol{y}_j\}$ is prohibitively difficult. A more feasible approach is to consider the average behavior of the assignment problem by assuming a Boltzmann distribution for the states of the system

$$P[\{V_{ij}\},\{\boldsymbol{y}_j\}] = e^{-E[\{V_{ij}\},\{\boldsymbol{y}_j\}]/T}/Z, \tag{59}$$

where the partition function Z is given by

$$Z = \sum_{\{V_{ij}\},\{\boldsymbol{y}_j\}} e^{-E[\{V_{ij}\},\{\boldsymbol{y}_j\}]/T}. \tag{60}$$

Summing over all allowed configurations of $\{V_{ij}\}$, the partition function becomes

$$Z = \sum_{\{\boldsymbol{y}_j\}} e^{-E_{\text{eff}}[\{\boldsymbol{y}_j\}]/T}, \tag{61}$$

with the effective energy

$$E_{\text{eff}}[\{\boldsymbol{y}_j\}] = -T\sum_i \ln\left(\sum_j e^{-|\boldsymbol{x}_i - \boldsymbol{y}_j|^2/T}\right) + \nu\sum_k |\boldsymbol{y}_k - \boldsymbol{y}_{k+1}|^2. \tag{62}$$

This is the same as the energy function of the elastic net method (Durbin and Willshaw, 1987). As the sought minimum configuration of the system dominates the be-

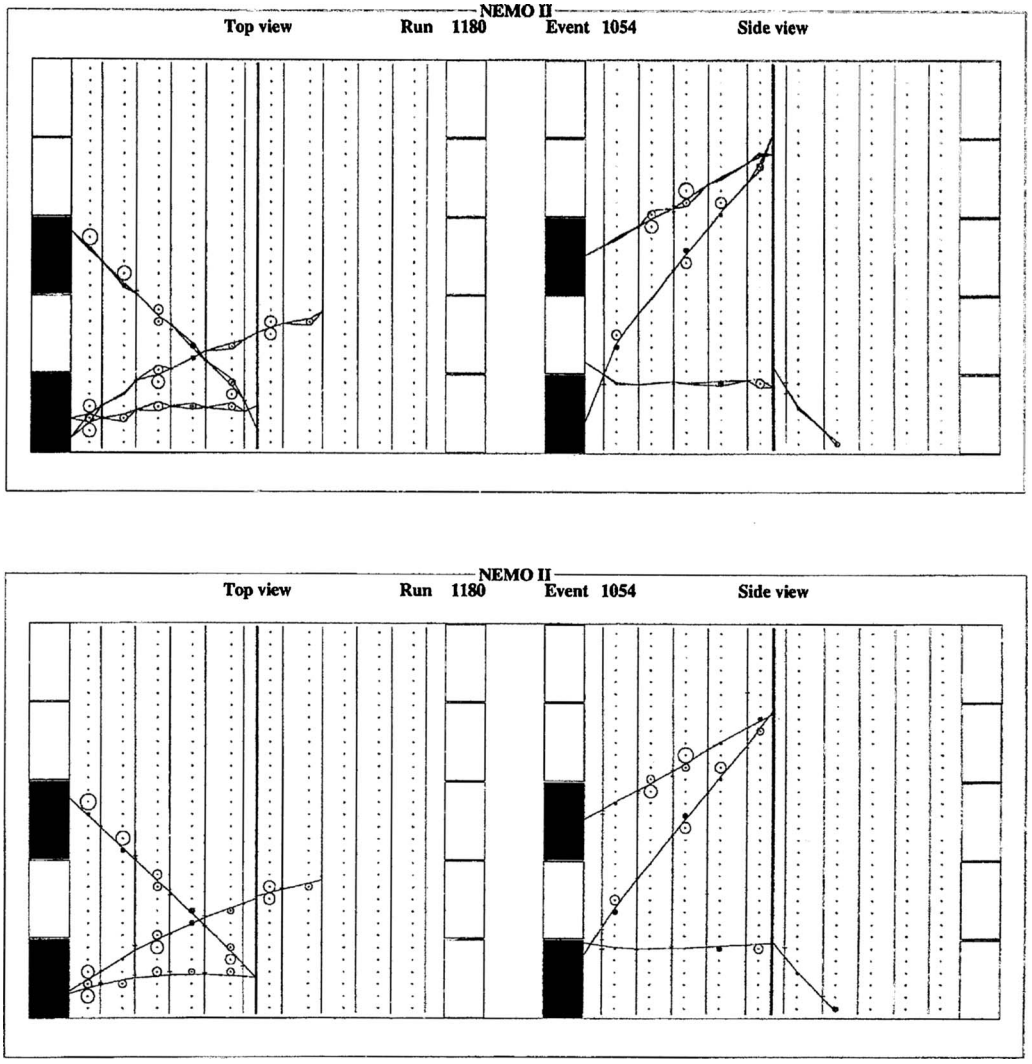

FIG. 11. Initial and final configurations of the elastic net for a NEMO-2 event. Some initial configurations are eliminated by grouping hits into clusters. From Kisel *et al.*, 1997.

havior of the partition function in the low-temperature limit, a feasible strategy is to initiate the search at a large temperature and find the minimum of the effective energy. The temperature is lowered, and a new minimum is found, starting out at the values of $\{\boldsymbol{y}_j\}$ found by the previous minimization. The procedure is repeated at successively lower temperatures, taking the zero-temperature limit in the end. This procedure is equivalent to the elastic net method iterated at successively lower values of the constant K (Durbin and Willshaw, 1987). With the introduction of a parameter interpreted as temperature, such a procedure can be regarded as an annealing procedure. Being nonrandom—in contrast to the stochastic strategy called simulated annealing (Kirkpatrick *et al.*, 1983)—it is an example of a deterministic annealing procedure.

The Hopfield network algorithm (Hopfield and Tank, 1985) can also be derived using Eq. (58) by considering the average behavior of $\{\boldsymbol{y}_j\}$ instead (Yuille, 1990). Such a strategy gives a new energy only dependent on $\{V_{ij}\}$, which is similar to the energy function of Hopfield and Tank (1985). This establishes the close connection between the elastic net method and the neural network approach.

The track reconstruction task [Ohlsson *et al.* (1992)] has been formulated as an assignment problem in a way similar to the approach described by Yuille (1990). In this so-called elastic arms approach, measurements i are assigned to template tracks or arms a by the binary decision units $\{S_{ia}\}$ under the constraint that each measurement is assigned to at most one arm a. The energy function reads

$$E[\{S_{ia}\},\{\boldsymbol{q}_a\}] = \sum_{ia} S_{ia} M_{ia} + \lambda \sum_i \left(\sum_a S_{ia} - 1 \right)^2, \tag{63}$$

where M_{ia} is the squared distance from measurement i to the track parametrized by the track parameter vector $\boldsymbol{q}_a$. The second term imposes a penalty if a measurement is not assigned to any track. Following exactly the same strategy as described for the traveling salesman problem above, the states of the system are assumed to obey a Boltzmann distribution, and an effective energy E_{eff} is

obtained by summing over all possible configurations of the assignment variables,

$$E_{\text{eff}}[\{\boldsymbol{q}_a\}] = -T\sum_i \ln\left(e^{-\lambda/T} + \sum_a e^{-M_{ia}/T}\right). \tag{64}$$

As before, the system is initiated at a large temperature with a certain set of values of the parameters of the arms. Successive minimizations are performed at a sequence of decreasing temperatures, stopping at a temperature close to zero. The minimization strategy at a given temperature is equivalent to the gradient descent procedure of Durbin and Willshaw (1987). Visually, this procedure amounts to a set of arms or template tracks being attracted to the measurements created by the charged particles during the annealing. At low temperatures, the arms settle in the vicinity of these measurements, and measurements too far from the arm have no effect on the final estimate of the track parameters due to the cutoff parameter λ.

The algorithm has been extended to deal with left-right ambiguities in drift-tube detectors (Blankenbecler, 1994), and improved minimization schemes using the Hessian matrices have been introduced (Ohlsson, 1993). The first test in an LHC scenario was done by Lindström (1995). Applied to track reconstruction in the transition radiation tracker (TRT) subsystem of the ATLAS detector (Aad *et al.*, 2008), the arms were initiated by the output of a Hough transform track finding procedure. The elastic arms algorithm (EAA) was then run in order to resolve the left-right ambiguities and settle on the final estimates of the track parameters.

A similar approach, called the elastic tracking algorithm (ETA), was developed by Gyulassy and Harlander (1991). The basic idea is to interpret the classical Radon transform as an interaction energy between the measurements in a tracking detector and a template track. The minimum of the interaction energy with respect to the parameters of the template track defines the solution to the problem. Similar to the elastic arm algorithm, an iterative annealing procedure is followed in order to avoid local minima of the energy function. A comparison between the Denby-Peterson network and the elastic tracking algorithm is shown in Fig. 12.

The interaction energy R_V as presented by Gyulassy and Harlander (1991) can in principle handle continuous charge distributions, i.e., situations where the charged particle potentially has ionized several neighboring basic detector units. Usually, the information from such neighboring units is compressed into an effective measurement before track reconstruction starts, implicitly discretizing the data. For discretized data the interaction energy is defined in terms of a sum over measurements i and tracks a,

$$R_V(\boldsymbol{q},I) = -\sum_{ia} V(M_{ia},I), \tag{65}$$

where V is an interaction potential and I is the iteration number. From Gyulassy and Harlander (1991) a Lorentzian potential of the following form was suggested:

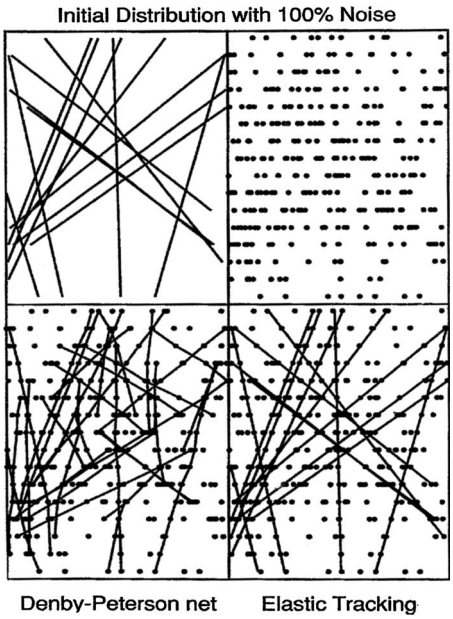

FIG. 12. A hard problem of tracking 15 lines on the top left given the measured points on top right including 100% noise. The confusion of the neural net algorithm (Denby, 1988; Peterson, 1989) on the bottom left is apparent. The long-range correlations are, however, easily identified with the elastic tracking method on the bottom right. From Gyulassy and Harlander, 1991.

$$V(M_{ia},I) = \frac{w^2(I)}{M_{ia} + w^2(I)}, \tag{66}$$

where w^2 is chosen quite large in the beginning of the iterations in order to smooth out the energy surface; asymptotically it should reach a value compatible with the standard deviation of the measurement error. The algorithm has been tested out in a time projection chamber-like scenario with nearby and overlapping tracks (Gyulassy and Harlander, 1992). Working on nonclustered data with realistic charge fluctuations, subpad multiple track resolutions were obtained.

A potential disadvantage with the deformable templates approach is the high computational complexity. An attempt to speed up the elastic arms algorithm was made by formulating it as a single-track algorithm (Frühwirth and Strandlie, 1999). At the same time, it was generalized in order to cope with an arbitrary number of competing measurements in the same detector layer. It was shown that despite the simplification to the single-track formulation, further computational load nevertheless had to be introduced in order to achieve optimal performance as advanced approaches such as quasi-Newton methods are needed in the final iteration for locating the global minimum of the energy function with satisfactory accuracy. Another disadvantage is the need of a parametric model of the track, which limits the application to experimental scenarios with vanishing or homogeneous magnetic fields, and negligible material effects such as energy loss and multiple scattering.

Track reconstruction and vertex reconstruction have many features in common. It is therefore not surprising that the deformable template method has been extended to the problem of vertex finding (Stepanov and Khanov, 1997). Simultaneous track and vertex finding with deformable templates was proposed by Haas (2000).

In the most general case, vertex finding starts with K tracks and N vertex templates. Each template is a collection of the following elements: the vertex position $\boldsymbol{v}_n$, the K binary decision units V_{nk}, the momentum vectors $\boldsymbol{p}_{nk}$ at vertex n of all tracks, and the estimated track parameters $\tilde{\boldsymbol{q}}_k$, where $1 \leqslant k \leqslant K$. The energy function reads

$$E[\{V_{nk}\},\{\boldsymbol{v}_n\},\{\boldsymbol{p}_{nk}\},\{\tilde{\boldsymbol{q}}_k\}] = \sum_{n=1}^{N}\sum_{k=1}^{K} V_{nk}M_{nk} + \lambda\sum_{k=1}^{K}\left(\sum_{n=1}^{N} V_{nk} - 1\right)^2, \tag{67}$$

where M_{nk} is a measure of distance between vertex template n and track k. The second term in Eq. (67) imposes a penalty when a track is not matched to exactly one vertex. The distance M_{nk} is chosen as the χ^2 statistic,

$$M_{nk} = \boldsymbol{r}_{nk}^{\mathrm{T}}\boldsymbol{G}_k\boldsymbol{r}_{nk}, \tag{68}$$

where $\boldsymbol{r}_{nk} = \tilde{\boldsymbol{q}}_k - \boldsymbol{h}_k(\boldsymbol{v}_n, \boldsymbol{p}_{nk})$ is the residual of track k with respect to vertex template n [see Eq. (28)] and $\boldsymbol{G}_k$ is the inverse covariance matrix of track k. If there is prior information that track k cannot be associated with vertex n, the corresponding decision unit V_{nk} can be frozen to 0.

Stepanov and Khanov (1997) minimized the energy function iteratively by alternating two steps. In the first one the energy function was minimized with respect to the continuous variables $\{\boldsymbol{v}_n\}$ and $\{\boldsymbol{p}_{nk}\}$ using gradient descent. In the second step, the weights were updated according to

$$V_{nk} = \frac{e^{-M_{nk}/T}}{e^{-\lambda/T} + \sum_{i=1}^{N} e^{-M_{ik}/T}}, \tag{69}$$

with a temperature parameter T. In the example presented by Stepanov and Khanov (1997), the starting temperature was chosen to be about 1 and slightly decreased during the iterations. The penalty term λ was chosen to be large (≈ 50) and slightly decreased as well in the course of the minimization.

C. Gaussian-sum filter

As seen in Sec. III.B, elastic tracking approaches can be used to resolve ambiguities and downweight noise. For instance, in the elastic arm approach the parameter λ plays the role as a cutoff parameter since measurements with a squared distance to the arm larger than λ have virtually no effect on the final estimates of the track parameters.

A first attempt to achieve a similar downweighting effect within a filtering approach took as a starting point a two-component Gaussian-mixture description of the measurement error, the narrow component representing true measurements and a wider component representing outliers (Guttman and Peña, 1985; Lin and Guttman, 1993; Frühwirth, 1995). The update of the track parameters and the covariance matrix is obtained in a Bayesian spirit as the mean and covariance matrix of the posterior distribution, resulting in a weighted sum of the output of two Kalman filters. This collapsed state vector is then propagated to the next detector layer containing a measurement.

The procedure was later, following Kitagawa (1989, 1994), generalized to a full Gaussian-sum filter (GSF) (Frühwirth, 1997). In the full GSF, both measurement noise and stochastic material effects such as multiple scattering and energy loss by bremsstrahlung are modeled as Gaussian mixtures. The posterior distribution is not collapsed after every update, and the GSF takes the form of a set of Kalman filters running in parallel. Each time a Gaussian-mixture probability density function with M components is encountered during the reconstruction procedure, the posterior distribution is still a Gaussian mixture but containing M times as many components as the prior distribution.

Let $\boldsymbol{x}$ denote the track state at a material layer or a measurement layer and let its prior distribution be a mixture of K multivariate Gaussians. The mixture density is completely specified by its weights, mean vectors, and covariance matrices,

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \varphi(\boldsymbol{x};\boldsymbol{x}_k,\boldsymbol{C}_k), \quad \sum_{k=1}^{K} \pi_k = 1. \tag{70}$$

If the state is at a material layer and the process noise (e.g., multiple scattering or bremsstrahlung) is modeled by a Gaussian mixture with M components,

$$g(\boldsymbol{x}) = \sum_{m=1}^{M} w_m \varphi(\boldsymbol{x};\boldsymbol{\mu}_m,\boldsymbol{Q}_m), \quad \sum_{m=1}^{M} w_m = 1, \tag{71}$$

the posterior density of the state vector is given by a mixture of all pairwise convolutions,

$$p(\boldsymbol{x}) = \sum_{k=1}^{K}\sum_{m=1}^{M} \pi_k w_m \varphi(\boldsymbol{x};\boldsymbol{x}_k + \boldsymbol{\mu}_m, \boldsymbol{C}_k + \boldsymbol{Q}_m). \tag{72}$$

If the state is at a measurement layer and the measurement error is modeled by a Gaussian mixture with M components,

$$h(\boldsymbol{m}) = \sum_{m=1}^{M} \omega_m \varphi(\boldsymbol{m};\boldsymbol{H}\boldsymbol{x} + \boldsymbol{c}_m, \boldsymbol{V}_m), \quad \sum_{m=1}^{M} \omega_m = 1, \tag{73}$$

the posterior density of the state vector is computed using Bayes' theorem,

$$p(\boldsymbol{x}) = \sum_{k=1}^{K}\sum_{m=1}^{M} p_{km} \varphi(\boldsymbol{x};\boldsymbol{x}_{km},\boldsymbol{C}_{km}), \tag{74}$$

with

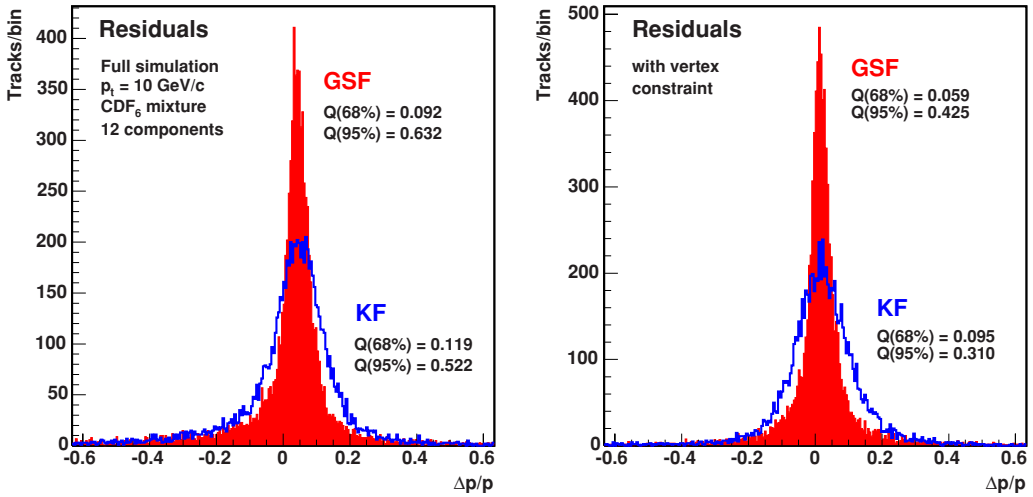

FIG. 13. (Color online) Normalized momentum residuals of electrons with (right) and without (left) the vertex constraint at $p_T = 10$ GeV/c. GSF (KF) results are shown as solid (open) histograms. The Gaussian-mixture approximation of the Bethe-Heitler model has six components. The maximum number of components in the GSF is 12. Histograms of 68% are contained in the respective intervals $[-Q(68\%), Q(68\%)]$; 95% of the histograms are contained in the respective intervals $[-Q(95\%), Q(95\%)]$. From Adam *et al.*, 2005.

$$\boldsymbol{x}_{km} = \boldsymbol{x}_k + \boldsymbol{C}_{km}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{V}_m^{-1}(\boldsymbol{m} - \boldsymbol{c}_m - \boldsymbol{H}\boldsymbol{x}_k),$$

$$\boldsymbol{C}_{km} = [\boldsymbol{C}_k^{-1} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{V}_m^{-1}\boldsymbol{H}]^{-1}, \tag{75}$$

$$p_{km} \propto \pi_k \omega_m \varphi(\boldsymbol{m}; \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{c}_m, \boldsymbol{V}_m + \boldsymbol{H}\boldsymbol{C}_k\boldsymbol{H}^{\mathrm{T}}).$$

The constant of proportionality is determined by the constraint

$$\sum_{k=1}^{K}\sum_{m=1}^{M} p_{km} = 1.$$

This procedure leads to an exponentially increasing number of components in the posterior distributions, and for practical purposes the maximum number of components has to be limited. Depending on the application, this can be achieved by keeping the N components with largest posterior weights or by merging components being close in parameter space, closeness defined by an appropriate similarity metric.

A GSF has been developed for the resolution of drift time ambiguities by modeling the measurements as a two-component mixture with different means, corresponding to the two potential positions of the measurement but equal standard deviations (Frühwirth and Strandlie, 1999). A potential drawback of this approach is the sensitivity to wrong or noisy measurements particularly in the early phases of the filter where the track parameters are poorly defined. Good initialization of the track parameters is therefore essential. The method has been made more robust by introducing the concept of a missing hit (Frühwirth and Strandlie, 2006), which implies that an additional component is created whenever a detector unit with a measurement is reached. This additional component is transported directly through the detector unit and corresponds to a measurement missing due to detector inefficiencies or being wrong, for example, because of track finding inefficiencies.

A totally different application of the GSF is the treatment of bremsstrahlung energy loss of electrons, which follows a highly non-Gaussian distribution. The Bethe-Heitler model (Bethe and Heitler, 1934) of this distribution has been approximated in terms of Gaussian mixtures for a range of radiation thicknesses (Frühwirth, 2003). A proof-of-principle study was reported by Frühwirth and Frühwirth-Schnatter (1998), and the first experimental application was made and tested out on simulated data in the CMS experiment (Adam *et al.*, 2005). A comparison of the GSF with the standard Kalman filter (KF) using electron tracks from the full CMS simulation, is shown in Fig. 13. The figure shows the distribution of the normalized difference between estimated and true momentum, both without and with a vertex constraint. The distribution of the GSF estimates has a significantly narrower core than the KF distribution, although the tails are slightly longer. The GSF was also implemented in the ATLAS experiment at the LHC and was shown, at least for simulated data, to improve the momentum resolution as compared to the standard Kalman-filter based algorithm (Aad *et al.*, 2008).

A related algorithm, the dynamic-noise-adjustment method (Kartvelishvili, 2007), was shown to yield similar results with less computational load. It can be regarded as a simplified version of the GSF. It makes a selection between a radiating and a nonradiating component at each material layer and thereby propagates only a single component through the detector.

Another application of the GSF has been found for the task of discriminating between different types of material encountered during a track reconstruction procedure (Strandlie and Frühwirth, 2006a). In a detector setup with complex structures consisting of different types of material, the standard approach is to create a

reconstruction geometry with averaged material properties. The GSF approach is to create one component for each of the different types of material present in the structure and using the measurements *a posteriori* to decide which of the types of material the particle actually went through.

If the vertex fit is formulated as a Kalman filter (see Sec. II.D.1), the latter can be generalized to a Gaussian-sum filter by taking a similar approach as in track fitting (Frühwirth and Speer, 2004; Speer and Frühwirth, 2006). As multiple scattering usually is absent in vertex fitting, it is the observations, i.e., the estimated track parameters of a previous track fit, which are modeled by Gaussian mixtures. The Gaussian-sum filter can be useful in two scenarios. In many cases the distribution of the estimation errors shows a Gaussian core and nonnegligible tails. These tails reflect imperfections of the track reconstruction. They may be caused by the track finder picking up wrong hits or distorted hits or by deficiencies in the material description, the calibration, or the alignment of the tracking device. In this first scenario, the components of the Gaussian in general have the same mean vector but different covariance matrices. In the second scenario, the track itself has been estimated by a GSF, for instance, because it is an electron track. In this second scenario, the components have different mean vectors and very likely also different covariance matrices. The most general mixture model for the estimated track parameters is therefore the following one:

$$f(\boldsymbol{q}) = \sum_{m=1}^{M} \omega_m \varphi(\boldsymbol{q};\boldsymbol{q}_m,\boldsymbol{V}_m), \quad \sum_{m=1}^{M} \omega_m = 1. \tag{76}$$

In the first scenario, the joint mean vector of all components is the actual estimate $\tilde{\boldsymbol{q}}$.

If a track is added to the vertex in the course of the GSF, the prior distribution of the vertex position is itself a Gaussian mixture,

$$g(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \varphi(\boldsymbol{v};\boldsymbol{v}_k,\boldsymbol{C}_k), \quad \sum_{k=1}^{K} \pi_k = 1. \tag{77}$$

After the track has been added, the joint posterior of the vertex position and the track momentum is a mixture with $M \cdot K$ components,

$$p(\boldsymbol{v},\boldsymbol{p}) = \sum_{k=1}^{K} \sum_{m=1}^{M} p_{km} \varphi((\boldsymbol{v},\boldsymbol{p});(\boldsymbol{v}_{km},\boldsymbol{p}_{km}),\Omega_{km}). \tag{78}$$

The calculation of the components weights p_{km}, the mean vectors $(\boldsymbol{v}_{km},\boldsymbol{p}_{km})$, and the joint covariance matrices Ω_{km} is analogous to Eq. (75). For more details about the limitation of the number of components, smoothing, and results with simulated data, see Speer and Frühwirth (2006).

D. EM algorithm and adaptive track fitting

In the single-track formulation of the elastic arm algorithm, the energy function E reads

$$E(\{S_k,s_{ik}\},\boldsymbol{q}) = \sum_k \left[S_k \left(\sum_{i=1}^{n_k} s_{ik} M_{ik} \right) + \lambda (S_k - 1)^2 \right], \tag{79}$$

where the sum over k denotes sum over layers, n_k is the number of measurements in layer k (including double counting due to potential left-right ambiguities in a drift-tube detector), and M_{ik} is the squared distance from measurement i in layer k to the single arm or track under consideration. The track parameter vector is denoted by $\boldsymbol{q}$. The binary assignment variables S_k and s_{ik} are either 0 or 1, and s_{ik} obeys the constraint $\Sigma_{i=1}^{n_k} s_{ik} = 1$. The variable S_k is 0 if all measurements in layer k are regarded as noise and 1 if one of them is not. The effective energy becomes in this case

$$E_{\text{eff}}(\boldsymbol{q}) = -T \sum_k \ln \left(n_k e^{-\lambda/T} + \sum_{i=1}^{n_k} e^{-M_{ik}/T} \right). \tag{80}$$

As mentioned, the minimization of this nonquadratic energy function in general requires time-consuming nonlinear approaches in order to locate the minimum with a required accuracy. It was realized by Ohlsson *et al.* (1992) and Yuille *et al.* (1994) that the EM algorithm is an alternative way of minimizing such energy functions. The EM algorithm proceeds by alternating expectation and minimization steps and is guaranteed to converge to a, possibly local, minimum of the energy function (Dempster *et al.*, 1977).

The expectation step calculates the average of the original energy function as given in Eq. (79) over the assignment variables, conditioned on the current value $\boldsymbol{q}'$ of the track parameters. This defines the function

$$Q(\boldsymbol{q}|\boldsymbol{q}') = \sum_{\{S_k,s_{ik}\}} E(\{S_k,s_{ik}\},\boldsymbol{q}) \cdot P(\{S_k,s_{ik}\}|\boldsymbol{q}'), \tag{81}$$

where $P(\{S_k,s_{ik}\}|\boldsymbol{q}')$ is the joint distribution of all assignment variables, given that the track parameters are equal to $\boldsymbol{q}'$. The result of the summation is (Frühwirth and Strandlie, 1999)

$$\begin{aligned} Q(\boldsymbol{q}|\boldsymbol{q}') &= \sum_k \left(\sum_{i=1}^{n_k} M_{ik} \frac{e^{-M'_{ik}/T}}{n_k e^{-\lambda/T} + \sum_{j=1}^{n_k} e^{-M'_{jk}/T}} \right. \\ &\quad \left. + \lambda \frac{n_k e^{-\lambda/T}}{n_k e^{-\lambda/T} + \sum_{j=1}^{n_k} e^{-M'_{jk}/T}} \right) \\ &= \sum_k \left(\sum_{i=1}^{n_k} M_{ik} p'_{ik} + \lambda p'_{0k} \right), \end{aligned} \tag{82}$$

where the primed quantities refer to a track with parameters $\boldsymbol{q}'$. The interpretation of p'_{ik} is the probability of

measurement i in layer k being assigned to the track, whereas p'_{0k} is the probability that none of the measurements is assigned to the track.

In the minimization step the function Q is minimized with respect to the parameters $\boldsymbol{q}$, regarding the primed quantities as fixed. The last term in Eq. (82) is independent of $\boldsymbol{q}$, so the same result is obtained if

$$Q_1(\boldsymbol{q}|\boldsymbol{q}') = \sum_k \sum_{i=1}^{n_k} M_{ik} p'_{ik} \tag{83}$$

is minimized with respect to the parameter vector $\boldsymbol{q}$. The new value of the parameter vector is used to recalculate the assignment probabilities, and the function Q_1 is again minimized. These alternating expectation and minimization steps are repeated until convergence, and the EM algorithm can be seen to amount to an iteratively reweighted least-squares procedure, where the weights are given by the assignment probabilities. The risk of reaching a local minimum of the energy function can again be decreased by annealing the temperature parameter T.

This is an important result. Through the application of the EM algorithm, the nonlinear problem of minimizing a nonquadratic energy function is solved by iterated minimizations of a quadratic energy function, which can be done by a linear least-squares estimator. If the Kalman filter is chosen as the linear least-squares estimator, the elastic arm algorithm becomes an iterated Kalman filter with annealing. The resulting algorithm is called the deterministic annealing filter (DAF) (Frühwirth and Strandlie, 1999). It has the additional advantage with respect to the standard formulation of the algorithm that material effects such as energy loss and multiple scattering can be taken into account in a straightforward manner. In addition, track models that do not have a simple parametric form can also be dealt with as the prediction step of the Kalman filter in general can consist of track propagation through an inhomogeneous magnetic field. An example of the outcome of a track fit using the DAF is shown in Fig. 14. In this example it is evident that annealing helps to avoid local minima of the energy function.

On the other hand, the running of a weighted Kalman filter as part of one iteration of the DAF can be regarded as a simplified version of a Gaussian-sum filter, where the GSF state vector is collapsed after every update. The iteration procedure can be motivated from a desire of overcoming the problem of insufficient information in the initial phase of the GSF. The DAF is thus seen to build a bridge between approaches based on neural networks or deformable templates and more classical approaches based on the application of stochastic filters.

With the GSF analogy in mind, the assignment weights of the DAF can be made more general than those arising from the elastic arm algorithm. In particular, they can take into account the fact that competing observations need not have the same precision. As a consequence they are similar to the GSF weights but do not include the covariance matrix of the predicted state since the track from the previous iteration is regarded as fixed during an expectation step of the EM algorithm,

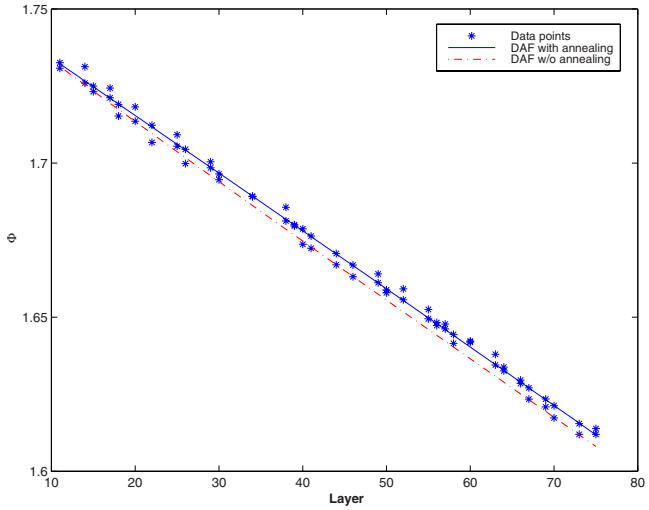

FIG. 14. (Color online) A track fitted with the DAF, with (solid line) and without (dash-dotted line) deterministic annealing. Without annealing the algorithm is seen to find a suboptimal solution to the optimization problem. Adapted from Strandlie and Zerubia, 1999.

$$p_{ik} = \frac{\varphi(\boldsymbol{m}_{ik};\boldsymbol{H}_k\boldsymbol{x}^*_{k|n},\boldsymbol{V}_{ik})}{c + \sum_{j=1}^{n_k} \varphi(\boldsymbol{m}_{jk};\boldsymbol{H}_k\boldsymbol{x}^*_{k|n},\boldsymbol{V}_{jk})}, \tag{84}$$

where φ is again the Gaussian probability density function, $\boldsymbol{m}_{ik}$ is measurement i in layer k, $\boldsymbol{V}_{ik}$ is the associated covariance matrix, $\boldsymbol{x}^*_{k|n}$ is the smoothed prediction from the previous iteration, and c is a cutoff term resembling the term containing λ in the elastic arm weights given in Eq. (82). A further generalization of the calculations of the assignment weights is presented by Strandlie and Zerubia (1999), allowing arbitrary probability distribution functions of the measurement errors.

The multitrack formulation of the elastic arm algorithm is via the EM algorithm generalized to a multitrack formulation of the DAF. This algorithm, called the multitrack filter (MTF) (Strandlie and Frühwirth, 2000), amounts to a set of iterated Kalman filters running in parallel (see Fig. 15). As for the DAF with respect to the single-track version of the elastic arm algorithm, decisive features are, e.g., the ability of taking material effects and inhomogeneous magnetic fields correctly into account.

The fact that the minimization of a nonquadratic energy function can be done by iterated minimizations of a quadratic function has quite far-reaching consequences. The minimization of the quadratic function can in principle be done by any relevant least-squares estimator. For instance, in a two-dimensional detector setup where the track model is the arc of a circle, a linear circle fit method based on a mapping to the Riemann sphere and fitting a plane to three-dimensional measurements leads

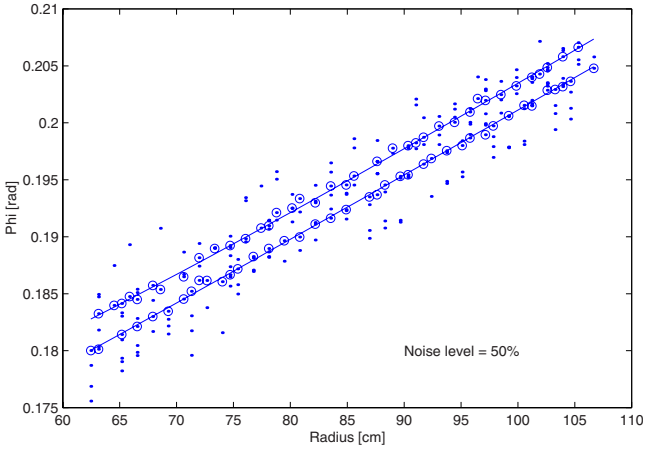

FIG. 15. (Color online) Two nearby tracks from a simulated event in the ATLAS TRT are fitted with the MTF. Since the TRT is a drift-tube detector, there are two ambiguous measurements for each tube hit by a charged particle. Additional noise hits are added in about every second tube. The true measurements are marked by open circles. The solid lines are the fitted tracks. From Strandlie and Frühwirth, 2000.

to an adaptive approach called the elastic planes algorithm (Strandlie *et al.*, 2000). Since the backbone method for fitting is a very fast approximate circle fit, the elastic planes algorithm shown by Strandlie *et al.* (2000) is much faster than the DAF and equally precise as long as the assumption of the purely circular track model holds at least approximately.

Another important consequence is that the DAF and the MTF immediately can be translated into analogous methods for adaptive, iterated, linear estimation of vertices, as first suggested by Strandlie (2000). Adaptive methods of vertex fitting are described in Sec. III.F.

E. Comparative studies

The DAF was first implemented for simulation studies in the barrel part of the ATLAS transition radiation tracker (Frühwirth and Strandlie, 1999). The barrel TRT consists of 75 layers of drift tubes (straws). The layers as well as the tubes in each layer are separated by about 6.8 mm. Because of the inherent left-right ambiguity of the drift time measurement, most hits have a mirror hit. The DAF was compared in terms of precision and running time to the following other methods: the GSF, the EAA, and the ETA. Two different annealing schedules were used for the DAF and EAA. In the "nominal schedule" the final value of the temperature T corresponded to the actual standard error of the observations, whereas in the "frozen schedule" the final value of T was smaller by a factor of 9. The result of the GSF was computed in two ways using either all components or only the best (most probable) component. The ETA was implemented using either a Gaussian (ETA-G) or a Lorentzian (ETA-L) potential. The results for tracks with mirror hits are shown in Table I. The precision of the method is quantified by the relative generalized variance (determinant of the covariance matrix) of the track parameters with respect to the true ones, the baseline being the DAF for tracks without mirror hits.

TABLE I. The relative generalized variance V_{rel} and the relative running time t_{rel} of various adaptive estimators for tracks with mirror hits. The baseline is the DAF for tracks without mirror hits. The different versions of the methods are explained in the text. Adapted from Frühwirth and Strandlie, 1999.

Method	V_{rel}	t_{rel}
DAF nominal	1.54	1.21
DAF frozen	1.74	1.41
GSF all	1.59	7.04
GSF best	1.78	7.04
EAA nominal	1.56	2.12
EAA frozen	1.71	2.44
ETA-G	3.11	2.38
ETA-L	3.51	2.87

Some interesting conclusions may be drawn from Table I. First, the mathematical equivalence of the EAA and the DAF is illustrated by nearly identical results. The DAF is, however, faster by almost a factor of 2. Second, cooling the DAF below the nominal standard deviation of the observations slightly deteriorates the precision. Third, using the full posterior information of the GSF is slightly better than just using the most probable component. The latter is equivalent to a combinatorial Kalman filter (see Sec. II.B.3). The GSF is considerably slower than the other methods. Finally, the ETA is about as fast as the EAA but not competitive in terms of precision.

The robustness of the estimators was studied by contaminating the tracks with noise. In each tube, the correct drift distance was replaced by a random drift distance with a probability of 10%. The results are shown in Table II.

It is evident that the performance of the GSF has deteriorated considerably, showing its inherent lack of robustness. The precision of the ETA is again worse than the one of the DAFs, the Gaussian potential being

TABLE II. The relative generalized variance V_{rel} and the relative running time t_{rel} of various adaptive estimators for tracks with mirror hits and noise. The baseline is the DAF for tracks without mirror hits. The different versions of the methods are explained in the text. Adapted from Frühwirth and Strandlie, 1999.

Method	V_{rel}	t_{rel}
DAF nominal	3.96	1.19
GSF all	27.33	6.86
ETA-G	5.77	2.72
ETA-L	6.56	2.89

TABLE III. The relative generalized variance of the multi-track fit with competition schemes 1–4 at various noise levels with mirror hits. The baseline is a single-track fit of all tracks separately. Reprinted from Strandlie and Frühwirth, 2000.

	Competition scheme			
Noise level (%)	1	2	3	4
0	281	36.2	4.52	2.84
10	270	58.7	5.35	4.35
20	388	100.9	6.26	7.06
30	358	185.3	7.19	9.51
40	409	238.9	9.50	12.44
50	653	301.6	11.66	17.65

somewhat less susceptible to the influence of the mirror and noise hits.

The MTF was first studied by Strandlie and Frühwirth (2000) using pairs of tracks in the ATLAS barrel TRT. A typical pair of tracks, including all hits, mirror hits and additional noise hits, is shown in Fig. 15. Four competition schemes were analyzed:

(1) *Competition between hits.* In each layer, there is competition between all hits for each track, but no competition between the tracks. This scheme is equivalent to the DAF.

(2) *Competition between tracks.* In each layer, there is competition between all tracks for each hit but no competition between the hits. This scheme is equivalent to the original formulation of the EAA (Ohlsson *et al.*, 1992).

(3) *Global competition.* In each layer, all tracks compete for all hits.

(4) *Competition between tracks and between mirror hits.* This is a refinement of scheme 2, specific for detectors with mirror hits.

With the MTF, the initial values of the track parameters play an important role. In order to separate the effect of the initialization from the properties of the competition schemes, the track parameters were initialized at their true values. Table III shows the relative generalized variance of the four schemes for tracks with mirror hits and additional noise hits at various noise levels. The baseline is a single-track fit of all tracks.

Scheme 1, the DAF, is consistently the least precise. Scheme 2 is also less precise than schemes 3 and 4. Schemes 3 and 4 are comparable but not entirely equivalent. At noise levels below 20%, scheme 4 is better, whereas at higher noise levels scheme 3 is better. The explanation is that scheme 4 implements a competition between a hit and its mirror hit, whereas scheme 3 treats all hits in a layer in the same manner. If the dominant source of ambiguity are mirror hits, scheme 4 is best equipped to deal with this situation. If, on the other hand, noise hits are the dominant source, scheme 3 is better, being designed to cope with additional noise hits.

Other studies, based on more detailed simulations, have been made in the CMS tracker (Winkler, 2002; Speer *et al.*, 2006b). For track reconstruction in dense jets, the DAF was shown to yield substantial improvements both in track parameter resolutions and in the overall track quality with respect to the standard Kalman-filter based procedure. The MTF provided resolutions similar to the DAF but gave better estimates of the track parameter covariance matrices in the most difficult situations, i.e., in the core of the jets with the highest energy.

The possibility of applying adaptive algorithms such as the DAF and the GSF for track reconstruction in scenarios with very large amounts of background and noise have been addressed in a couple of feasibility studies (Frühwirth and Strandlie, 2006; Strandlie and Frühwirth, 2006b). In these studies the robust version of the GSF, allowing for missing observations, was used.

Two different experimental setups were studied. One of them (setup C) is an all-silicon tracking detector similar to the CMS tracker at the LHC. The other one (setup A) is a combined silicon and drift-tube detector system similar to the ATLAS inner detector at the LHC. In both experiments, track finding was starting from seeds that were track segments in the innermost parts of the detectors. These seeds consisted of measurements in three consecutive layers with pixel resolution in setup C, whereas the seeds consisted of measurements from seven consecutive layers (three with pixel resolution and four with silicon strip resolution) in setup A. Track finding was then carried out in the 10 outermost silicon strip layers in setup C and in the 36 drift-tube layers in setup A. In both cases the track fit was carried out using the full information from all measurements in the track candidates, including the seeds.

Results from the simulation experiments with setup C, similar to the CMS tracker, are shown in Fig. 16. Track finding was performed using either the combinatorial Kalman filter (CKF) or the GSF, whereas the track fit was done either by a standard Kalman smoother (KSM) or by the DAF. In addition to the correct measurements, noise was added with a density of 3 hits/mm^2 in the innermost silicon layer, decreasing inversely proportional to the radius. In addition, the true hits were replaced by noise hits with a 30% probability. The results are shown as a function of the quality of the measurements constituting the seeds. The pixel hits were distorted by increasing the width of the Gaussian smearing with a scale factor k. Thus $k=1$ gives perfect seeds, and increasing the value of k gives seeds which are gradually deteriorated.

The efficiency of track finding is seen to be quite similar for all combinations of track finding and track fitting methods. As expected, the performance gets worse as the quality of the seeds decreases. The resolution is consistently, but not dramatically, better using the DAF in the track fit as compared to the KSM. Track finding at a noise level of 10 hits/mm^2 in the innermost silicon layer was also studied (Frühwirth and Strandlie, 2006), but at this noise level track finding broke down completely be-

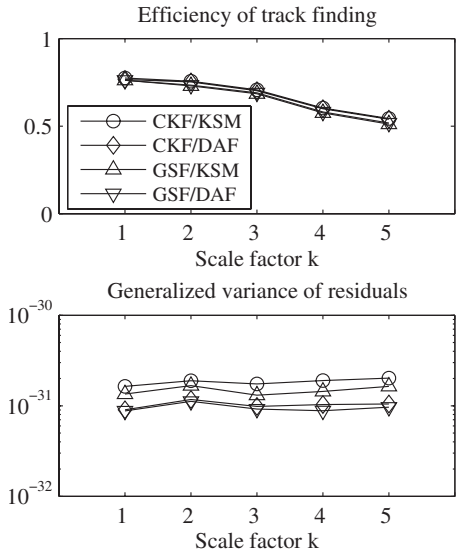

FIG. 16. Track finding efficiency and track fitting accuracy—expressed by the generalized variance of the residuals of the fitted track parameters with respect to the true values of the parameters—for various combinations of track finding and track fitting methods are shown as a function of the scale factor k. The results are from setup C, similar to the CMS tracker. From Strandlie and Frühwirth, 2006b.

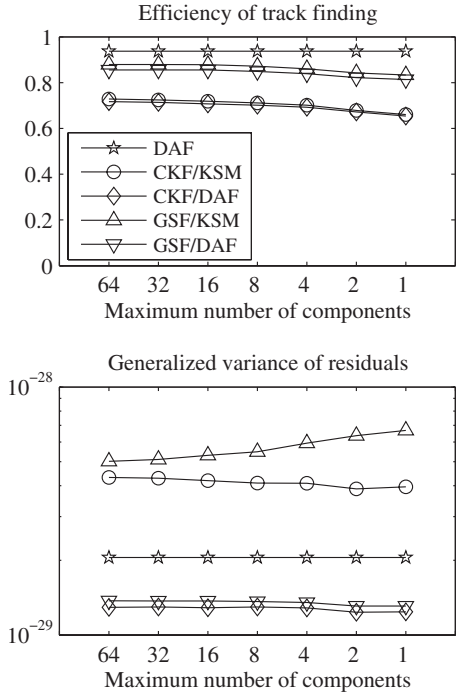

FIG. 17. Track finding efficiency and track fitting accuracy—expressed by the generalized variance of the residuals of the fitted track parameters with respect to the true values of the parameters—for various combinations of track finding and track fitting methods are shown as a function of the maximum number of components kept by the CKF or the GSF. The results are from setup A similar to the ATLAS inner detector. From Strandlie and Frühwirth, 2006b.

cause almost all track candidates found were just random collections of noise hits.

Results from the simulation experiments with setup A, similar to the ATLAS inner detector, are shown in Fig. 17. The performance is shown as a function of the maximum number of components or branches kept by the GSF or the CKF. In addition to the same combinations of track finding and track fitting methods as shown in Fig. 16, a fully DAF-based combined track finding and fitting was tried out. In this case, the seed was extrapolated through the entire volume of the drift-tube detector, and measurements in a band along the extrapolated track constituted the track candidate used by the DAF track fit.

It can be seen that the track finding efficiency of the GSF is better than that of the CKF and that the accuracy of the DAF track fit is better than the KSM track fit. The efficiency of the stand-alone DAF is by far the best in this case, as it finds several of the tracks not being found by the CKF or GSF due to too many missing hits. The difference in accuracy between the stand-alone DAF and the DAF track fit initiated by the CKF or GSF track finding is also due to the larger track sample fitted by the stand-alone DAF, containing tracks with more missing hits and thereby yielding less estimation accuracy.

The stand-alone DAF was also tried out in setup C but gave a significantly smaller efficiency than the other methods. The reason is the larger amount of material present in a silicon-based detector as compared to a gaseous one, leading to more multiple scattering and the lower quality of the seeds. The first extrapolation through the entire detector can then be quite far away from the real track, and with a lot of noise hits present, the DAF is not able to recover.

In general, the stand-alone DAF seems to work best as compared to other methods when initiated with high-quality seeds and when used in detector setups with relatively little material. This can be explained by the fact that the DAF uses information from the entire set of measurements in the track candidate when calculating the assignment weights, and the full power of this feature is utilized in situations with long-range propagation of information from the measurements. With a significant amount of material present, the information from measurements far away is blurred by multiple scattering, and the assignment power decreases. In the track fit, however, the DAF seems at these noise levels to be the method of choice in both experimental setups.

F. Adaptive vertex fitting

The deformable templates method of vertex finding can be rewritten as an iterated reweighted least-squares estimator. In the case of a single vertex to be found, the algorithm is called the adaptive vertex fit (AVF) (Waltenberger, 2004; Waltenberger, Frühwirth, and Vanlaer, 2007). The estimator can be interpreted both as an EM algorithm and as a redescending M-estimator (Frühwirth and Waltenberger, 2008). If the estimator is formulated as an adaptive filter, the corresponding dynamical sys-

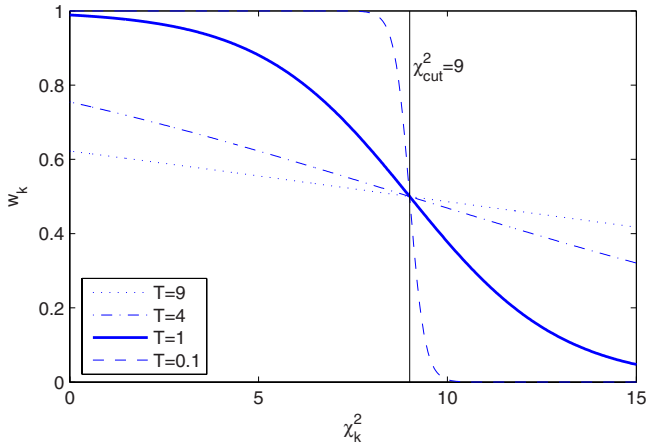

FIG. 18. (Color online) Adaptive vertex fit. The weight function w_k of track k as a function of the distance χ^2_k to the vertex for four values of the temperature. The cutoff is at $\chi^2_{\text{cut}}=9$.

tem has no process noise. In the light of the results in Sec. III.E it can therefore be expected that the adaptive approach is particularly appropriate to vertex fitting.

The AVF is an alternating sequence of two steps: estimation of the vertex position and computation of the track weights. If the estimator is enhanced by deterministic annealing, the temperature parameter is decreased after every iteration. In the EM interpretation, the weight of a track is the posterior probability of the track actually belonging to the vertex, called the assignment probability.

The estimation can be performed by a Kalman filter or by any other least-squares estimator, e.g., one of the simplified estimators described in Sec. II.D. The computation of the weight of track k is a special case of Eq. (69),

$$w_k = \frac{e^{-\chi^2_k/2T}}{e^{-\chi^2_{\text{cut}}/2T} + e^{-\chi^2_k/2T}}, \tag{85}$$

where χ^2_k is the weighted distance of track k from the current vertex position, χ^2_{cut} is a cutoff, and T is the temperature parameter. The shape of the weight function for a cutoff of $\chi^2_{\text{cut}}=9$ and different values of the temperature is shown in Fig. 18. The weight is equal to 0.5 if $\chi^2=\chi^2_{\text{cut}}$. At low temperatures the weight function approaches a step function, resulting in a yes or no decision about the assignment of the track to the vertex.

As mentioned, the AVF can also be interpreted as a redescending M-estimator with annealing. In this interpretation the estimator can be analyzed in terms of its influence function and associated concepts such as gross-error sensitivity and rejection point (Hampel *et al.*, 1986).

The weight function drops to zero very quickly for large values of χ^2_k. This poses a problem if the initial position of the vertex is very far from the true vertex position. Some care has to be taken to ensure that the initial vertex is where the majority of the tracks are. Various robust estimators with high breakdown point [see Huber (1981)] have been studied by Waltenberger (2004) and Waltenberger, Frühwirth, and Vanlaer (2007). An estimator has a high breakdown point if a large fraction of the data can be outliers without destroying the estimator. For a formal definition, see Rousseeuw and Leroy (1987).

Alternatively, hierarchical cluster finding methods as well as nonhierarchical methods such as vector quantization, the k-means algorithm, and deterministic annealing can be used to construct the initial vertex position (Chabanat *et al.*, 2005).

The AVF has originally been developed for the CMS experiment. It has also been used by the H1 experiment (Aktas *et al.*, 2005, 2006; Erdmann, 2006) and is one of the methods implemented for the ATLAS experiment (Piacquadio *et al.*, 2008). For comparisons to other vertex estimation methods, see Frühwirth *et al.* (2003a, 2003b), D'Hondt *et al.* (2004), and Speer *et al.* (2006a). As an example, Table IV shows a comparison between the AVF and the Kalman filter (KF) on two kinds of vertices: the primary vertex of the process $pp\to t\bar{t}H$ and the decay vertex of the B_s meson in the decay channel $B_s\to J/\psi\phi\to\mu^+\mu^-K^+K^-$. The data come from a full simulation in the CMS detector (Speer *et al.*, 2006a).

Table IV shows the average χ^2 of the fit divided by the number of degrees of freedom, the average calculation time, and three quantities that characterize the distribution of the residuals in the transverse (x) and longitudi-

TABLE IV. Comparison of the main properties of the Kalman filter (KF) and the AVF on two physics processes. Top: $B_s\to J/\psi\phi\to\mu^+\mu^-K^+K^-$. Bottom: $pp\to t\bar{t}H$. Reprinted from Speer *et al.*, 2006a.

$B_s\to J/\psi\phi\to\mu^+\mu^-K^+K^-$		
Filter	KF	AVF
Avg. χ^2/ndf	1.32	0.97
Avg. time (ms)	1.2	3.8
Res. x (μm)	55	54
95% cov. x (μm)	164	155
Pull x	1.08	1.02
Res. z (μm)	74	73
95% cov. z (μm)	471	440
Pull z	1.08	1.02
$pp\to t\bar{t}H$		
Filter	KF	AVF
Avg. χ^2/ndf	2.05	0.77
Avg. time (ms)	13	54
Res. x (μm)	14	10
95% cov. x (μm)	118	21
Pull x	1.51	0.99
Res. z (μm)	18	13
95% cov. z (μm)	122	30
Pull z	1.46	1.00

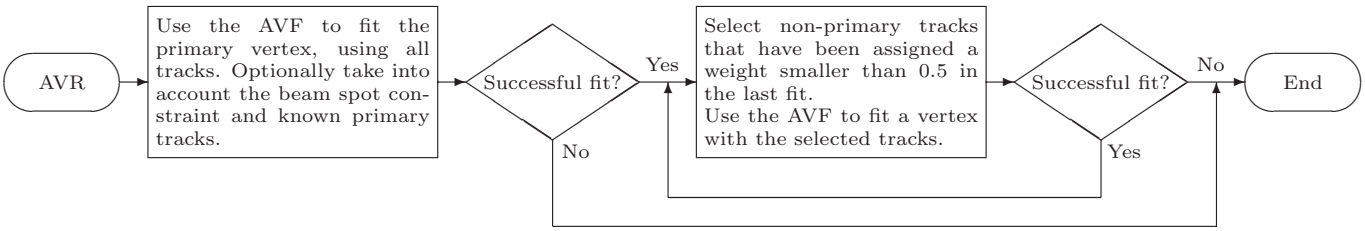

FIG. 19. Flow chart of the AVR based on the AVF. Adapted from Waltenberger, 2008.

nal (z) directions: the standard deviation of a Gaussian fitted to the distribution, the half-width of the symmetric interval covering 95% of the distribution, and the standard deviation of a Gaussian fitted to the standardized residuals (pulls).

In the low-multiplicity decay of the B_s the AVF is only slightly better than the KF. In the high-multiplicity primary vertex of the $t\bar{t}H$ events the AVF improves the resolution by about 30% and reduces the tails significantly. The estimated error, as indicated by the standard deviation of the pulls, is also much improved. Of course, the AVF is always slower than the KF.

The AVF can also be used to construct a general purpose vertex finder, the adaptive vertex reconstructor (AVR) (Waltenberger, 2008). Concisely stated, the AVR is an iterated AVF. The flow chart in Fig. 19 defines the algorithm in more detail. A study of its performance in the context of b tagging in CMS is reported by Waltenberger (2008).

The single-vertex AVF can be extended without difficulty to a multivertex fitter (MVF) (Frühwirth and Waltenberger, 2004) with $N>1$ vertices. The vertices now compete for the tracks. Consequently, the weight of track k with respect to vertex n is computed by

$$w_{nk} = \frac{e^{-\chi^2_{nk}/2T}}{e^{-\chi^2_{\text{cut}}/2T} + \sum_{i=1}^{N} e^{-\chi^2_{ik}/2T}}, \tag{86}$$

where χ^2_{ik} is the weighted distance of track k from the current position of vertex i, χ^2_{cut} is the cutoff, and T is the temperature parameter [see also Eq. (69)]. If there is a competing vertex nearby, the weight function of a track changes drastically as compared to the AVF. Figure 20 shows the weight w_{nk} of a track k with respect to a vertex n if there is a competing vertex i at a distance of $\chi^2_{ik}=1$. Even for $\chi^2_{nk}=0$ the weight w_{nk} is now only about 0.6 at a temperature $T=1$. It is only at very small temperatures that the algorithm decides unambiguously in favor of the closer vertex.

The MVF is implemented in the vertex reconstruction software of CMS and ATLAS (Costa, 2007). The adaptive algorithms described in this section are also available in the experiment independent vertex reconstruction toolbox RAVE (Waltenberger and Moser, 2006; Waltenberger, Mitaroff, and Moser, 2007).

IV. DETECTOR ALIGNMENT

A. Introduction

The models used for the estimation of track parameters [see Eq. (12)] and vertex parameters [see Eq. (28)] comprise a deterministic part, describing the motion of a charged particle in a magnetic field, and a stochastic part, describing the observation errors and the interaction of a charged particle with the material of the detector. There is, however, an additional source of uncertainty not taken into account so far, namely, the limited knowledge of the precise location and orientation of the sensitive elements of the detector. Determining the location and orientation to the required precision is called detector alignment. In some cases deformations of the sensitive elements, such as sagging or bending, also have to be determined.

There are various possibilities for the treatment of alignment corrections, ranging from simple translations and rotations, equivalent to those of a rigid body, to more complex deformations such as bends or twists. In a solid-state detector such as a silicon tracker the frequency of realignment depends mainly on the mechanical stability, which in turn depends on the temperature and on the magnetic field. In a gaseous detector such as a time projection chamber (TPC), tracks may be distorted not only because of a mechanical deformation but

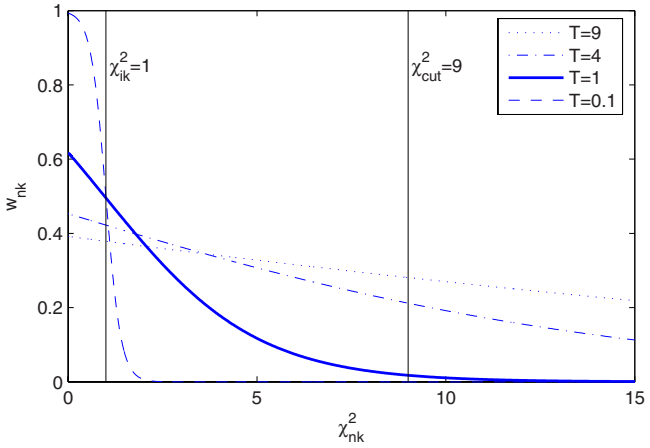

FIG. 20. (Color online) Multivertex fit. The weight function w_{nk} of track k with respect to vertex n as a function of the distance χ^2_{nk} to vertex n, if there is a competing vertex i at a distance of $\chi^2_{ik}=1$, for four values of the temperature. The cutoff is at $\chi^2_{\text{cut}}=9$.

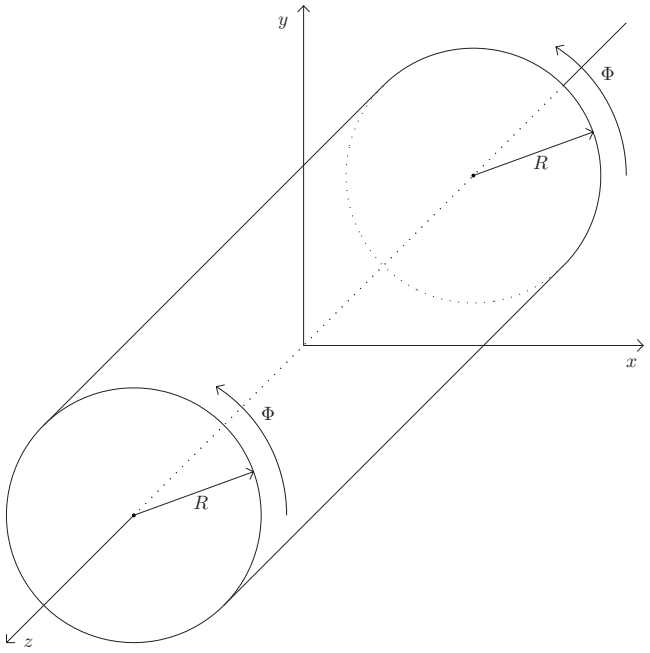

FIG. 21. A cylindrical detector and the coordinates R, Φ, and z. The effects of distortions in these three coordinates are described in Table V. For instance, a distortion $\Delta\Phi$ that varies with z is a twist.

because of inhomogeneities of the magnetic field in the chamber and the electric field used to operate the chamber (Dydak, 2003). As the buildup of positive ions contributes to the latter, the frequency of realignment depends also on the stability of the beam conditions (intensity and momentum).

Misalignment compromises tracking and vertex finding [see, e.g., Barbone *et al.* (2006) and Vanlaer *et al.* (2006)] and thus directly affects physics measurements such as momentum and invariant mass (Lampén *et al.*, 2008). It also deteriorates the precision of impact parameters and thereby reduces the efficiency of b-tagging algorithms (Lampén *et al.*, 2008). Alignment is therefore an extremely important task. As a general rule, the precision of the alignment should be significantly better than the intrinsic resolution of the sensitive elements. In order to achieve this, various strategies can be used. Sensor positions can be measured in the laboratory or *in situ* by lasers. To obtain the ultimate precision, however, reconstructed tracks have to be used.

With a sufficiently large number of tracks the statistical errors of the estimated alignment parameters can be made as small as required. The challenge is to control the systematic errors to the required level (Brown *et al.*, 2009). This is due to the fact that for any kind of tracks there are several degrees of freedom that are not constrained, usually referred to as weak modes, weakly defined modes, or χ^2-invariant modes. The most important global distortions of a detector with cylindrical symmetry (see Fig. 21) have been classified by Brown *et al.* (2007) and are reproduced in Table V.

A reasonable strategy to avoid global deformations is to use different kinds of tracks for track-based alignment, for instance, tracks from interactions, cosmic muons, and beam halo muons. An obvious weak mode is a translation or rotation of an entire tracking device, which can only be fixed with some kind of reference frame, be it an external system or by definition.

TABLE V. Main systematic distortions in a system with cylindrical geometry and multiple layers. Distortions in R, Φ, and z are considered as a function of these coordinates. The potential impact of these distortions on physics results is indicated (in italics). Adapted from Brown *et al.*, 2007.

	ΔR	$\Delta\Phi$	Δz
vs	Radial expansion	Curl	Telescope
R	*Distance scale*	*Charge asymmetry*	*z momentum*
vs	Elliptical	Clamshell	Skew
Φ	*Vertex mass*	*Vertex displacement*	*z momentum*
vs	Bowing	Twist	z expansion
z	*Total momentum*	*CP violation*	*Distance scale*

In many cases alignment algorithms were developed *ad hoc* and consequently were difficult to port to other experiments. The ever increasing requirements on alignment performance, both in terms of precision and in terms of the sheer number of parameters to be estimated, have instigated the development of a generic algorithm called Millepede (Blobel and Kleinwort, 2002; Blobel, 2006, 2007). Millepede has been used and is being used by several large experiments. It is worth noting that the largest LHC experiments, ATLAS and CMS (see below), which have the most difficult alignment tasks, do not rely on a single alignment method but have implemented several methods. This is extremely useful for debugging and cross validation.

The alignment of large tracking devices poses a serious computing challenge. It will be shown that it is possible to parallelize at least part of the alignment task. With a relatively small number of processors, of the order of 50, close to 10^5 alignment constants can be computed in less than 12 h (see Sec. IV.D.4.e).

B. Global alignment

The foundation of a track-based alignment method is an enhanced track model in which the observations depend not only on the track parameters $\boldsymbol{q}$ but also on a set of alignment parameters $\boldsymbol{a}$,

$$\boldsymbol{m} = \boldsymbol{d}(\boldsymbol{a},\boldsymbol{q}) + \boldsymbol{\gamma}, \quad \mathrm{var}(\boldsymbol{\gamma}) = \boldsymbol{V}, \tag{87}$$

where $\boldsymbol{a}$ describes the deviation of the ideal geometry from the actual geometry. For a sample of n tracks, the model can be written as

$$\begin{pmatrix} \boldsymbol{m}_1 \\ \vdots \\ \boldsymbol{m}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{d}_1(\boldsymbol{a},\boldsymbol{q}_1) \\ \vdots \\ \boldsymbol{d}_n(\boldsymbol{a},\boldsymbol{q}_1) \end{pmatrix} + \begin{pmatrix} \boldsymbol{\gamma}_1 \\ \vdots \\ \boldsymbol{\gamma}_n \end{pmatrix}. \tag{88}$$

It is evident that model (88) is formally the same as model (28) used in vertex estimation. Starting from approximate parameter values $\check{\boldsymbol{a}}$ and $\check{\boldsymbol{q}}_i$, model (88) can be expanded into an approximate linear model

$$\begin{pmatrix} \boldsymbol{m}_1' \\ \boldsymbol{m}_2' \\ \vdots \\ \boldsymbol{m}_n' \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{A}_2 & \boldsymbol{0} & \boldsymbol{B}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_n & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{B}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{q}_1 \\ \vdots \\ \boldsymbol{q}_n \end{pmatrix} + \begin{pmatrix} \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \\ \vdots \\ \boldsymbol{\gamma}_n \end{pmatrix}, \tag{89}$$

with

$$\boldsymbol{A}_i = \left.\frac{\partial \boldsymbol{d}_i(\boldsymbol{a},\boldsymbol{q}_i)}{\partial \boldsymbol{a}}\right|_{\check{\boldsymbol{a}},\check{\boldsymbol{q}}_i}, \quad \boldsymbol{B}_i = \left.\frac{\partial \boldsymbol{d}_i(\boldsymbol{a},\boldsymbol{q}_i)}{\partial \boldsymbol{q}_i}\right|_{\check{\boldsymbol{a}},\check{\boldsymbol{q}}_i},$$
$$\boldsymbol{m}_i' = \boldsymbol{m}_i - \boldsymbol{d}_i(\check{\boldsymbol{a}},\check{\boldsymbol{q}}_i) + \boldsymbol{A}_i\check{\boldsymbol{a}} + \boldsymbol{B}_i\check{\boldsymbol{q}}_i. \tag{90}$$

The main difference between vertex estimation and alignment is the fact that the vector $\boldsymbol{a}$ of global alignment parameters can be much larger than the vector $\boldsymbol{v}$ of global vertex parameters. Computing the estimated alignment parameters $\tilde{\boldsymbol{a}}$ according to Eq. (33) is therefore not always feasible. In general, however, it is sufficient to solve the system of linear equations,

$$\left(\sum_{i=1}^{n} \boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{G}_i^{\mathrm{B}}\boldsymbol{A}_i\right)\tilde{\boldsymbol{a}} = \sum_{i=1}^{n} \boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{G}_i^{\mathrm{B}}\boldsymbol{m}_i', \tag{91}$$

where $\boldsymbol{G}_i^{\mathrm{B}}$ is defined as in Eq. (33). This approach is taken in the Millepede program, originally developed for the alignment of the H1 experiment (Blobel and Kleinwort, 2002; Blobel, 2006, 2007) using advanced numerical methods to solve the system of linear equations. Millepede is written in FORTRAN but has been translated to C++ for the alignment of the LHCb vertex locator (VELO) (Viret *et al.*, 2005).

As in the case of the vertex fit, the alignment parameters can also be estimated by a Kalman filter (Frühwirth *et al.*, 2003; Widl *et al.*, 2006; Widl and Frühwirth, 2008). In this approach the alignment parameters are updated after a track is processed. In principle, all alignment parameters can be updated, but if their number is large, the update is restricted to the detector modules that have significant correlations with the ones in the current track. This is accomplished by the computation of a suitable measure of distance between modules based on recording which modules have been hit by the same track. This bookkeeping task consumes a significant part of the computing time required by the algorithm. The method can be parallelized by splitting the track sample into several subsamples and processing the subsamples on individual cores. The final alignment parameters are computed as the weighted mean of the individual results (Widl and Frühwirth, 2009).

C. Iterative alignment

Alternatively, the alignment parameters can be considered as missing data. The estimation is performed by an EM-like algorithm by iterating the following two steps: M step, reconstruct a sample of tracks using the current alignment parameters; E step, update the alignment parameters using the residuals of the reconstructed tracks.

The M step involves only estimation of track parameters using some standard estimation method and the calculation of residuals in the alignable modules crossed by the track. The E step can be accomplished in several ways. The simplest one is to fill the residuals into histograms and to extract the alignment corrections from the mean or median of the histograms. This approach is, however, restricted to those parameters which are directly accessible from the histograms, i.e., shifts. A more sophisticated approach is the parametrization of the residuals as a function of the track parameters. By fitting the parametrized models to the residuals, shifts and rotations can be extracted from the histograms. The statistically most satisfactory method is the minimization of a χ^2 function (Karimäki *et al.*, 2003) in each module to be aligned. For a given module, the χ^2 function has the form

$$\chi^2 = \sum_j \boldsymbol{\epsilon}_j^{\mathrm{T}}\boldsymbol{G}_j\boldsymbol{\epsilon}, \tag{92}$$

where μ is the residual between the observed position and the alignment dependent impact point of the track and $\boldsymbol{G}_j$ is the inverse covariance matrix of the observation. The sum is taken over all tracks seen by the module. The correction to the local alignment parameters $\boldsymbol{a}$ can be computed explicitly by

$$\delta\boldsymbol{a} = \left(\sum_j \boldsymbol{J}_j^{\mathrm{T}}\boldsymbol{G}_j\boldsymbol{J}_j\right)^{-1}\left(\sum_j \boldsymbol{J}_j^{\mathrm{T}}\boldsymbol{G}_j\boldsymbol{\epsilon}_j\right), \tag{93}$$

where $\boldsymbol{J}_j$ is the Jacobian matrix,

$$\boldsymbol{J}_j = \frac{\partial\boldsymbol{\epsilon}(\boldsymbol{a})}{\partial\boldsymbol{a}}, \tag{94}$$

computed at the current value of the local alignment parameters. The method is an essentially local one as each module is aligned independently from all the other ones.

D. Experimental examples

In the LEP era, the main task was the alignment of the various mostly gaseous tracking devices relative to a reference tracking detector [see, for instance, Wiedenmann (1992) and Andreazza and Piotto (1999)]. With the advent of large-scale semiconductor tracking devices the demands on the alignment methods have risen enormously both in terms of precision and in terms of the number of elements to be aligned. For instance, in the silicon vertex tracker (SVT) of BaBar there are 340 elements to be aligned (Brown *et al.*, 2009); in the central

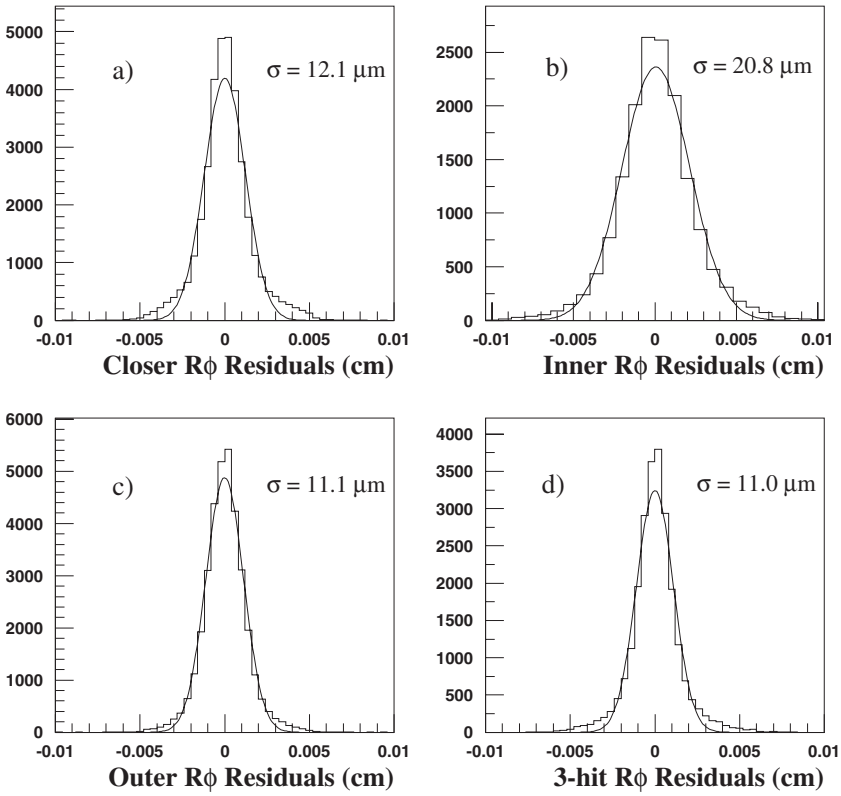

FIG. 22. Residual distributions after the internal alignment of the DELPHI VD for 1999 data taking: (a) tracks in closer overlaps, (b) tracks in inner overlaps, and (c) tracks in outer overlaps. The residuals are computed between a hit in the overlap and the track when it is forced through the other hit in the overlap. The width of each distribution has to be divided by $\sqrt{2}$ to obtain the single hit precision for that layer. Also shown are (d) the inner layer residuals for three-hit tracks. The tracks are forced through hits associated in closer and outer and interpolated to inner modules. The residuals are plotted between interpolated tracks and hits in inner modules. For this plot, the width has to be divided by $\sqrt{1.5}$ to obtain the silicon precision, assuming that it is the same in each of the three layers. From Andreazza and Piotto, 1999.

detector of the DØ experiment 976 elements (Sopczak, 2006); in the silicon tracker of the CDF experiment for Run IIb 2304 elements (Akimoto *et al.*, 2006); in the ATLAS silicon tracker 5832 elements (Aad *et al.*, 2008); and in the CMS inner tracker 13 300 elements (Chatrchyan *et al.*, 2008).

In order to address the challenges of detector alignment at LHC, a workshop was held in 2006. The proceedings (Blusk *et al.*, 2007) are a useful source of information, offering both experience from running experiments and describing the strategies of the four LHC experiments, including alignment validation (Golling, 2007). Follow-up workshops were held in 2007 and 2009. The presentations are available online (CERN Indico, 2007, 2009). Information about experiments at LEP and SLD was gathered from the, sometimes scant, literature.

1. *Z* factories

Track-based alignment in experiments at *Z* factories (LEP at CERN and SLD at SLAC) is characterized by the following traits: clean events with relatively small multiplicities; an abundance of high-energy muons; a precise knowledge of the center-of-mass energy of the collisions; and, as a consequence, the possibility to impose momentum constraints on the muons used for alignment.

a. DELPHI

At the LEP collider, the experiment with the most complex tracking system was DELPHI (Abreu *et al.*, 1996). In the barrel region it comprised three gaseous detectors—the central TPC, an inner drift (ID) chamber, and an outer drift (OD) chamber—and a silicon vertex detector (VD). In the forward region there were two forward drift chambers on each side, forward chamber A (FCA) and forward chamber B (FCB). In addition, the ring imaging Čerenkov counters and the muon chambers had to be aligned relative to the central tracking system. The alignment strategy of the DELPHI tracking detectors is described by Andreazza and Piotto (1999).

The OD was used to fix the z axis and the transverse coordinates, while the TPC was used to fix the origin of the coordinate system. The alignment procedure started with the internal alignment of the VD. This was obtained by assigning the status of master to one of the three layers (closer, inner, and outer). The master layer was aligned as a full pseudocylindrical object using all overlap constraint between adjacent modules. The modules of the other two layers were aligned individually with respect to the corresponding master modules. An example of the precision achieved in the internal alignment with 1999 data is shown in Fig. 22. For the internal alignment of the upgraded VD at LEP2, see Brückman de Renstrom (2004).

After the internal alignment, the VD was aligned globally, treating it as a rigid object. The transverse position was determined relative to the OD and the longitudinal position relative to the TPC. Once the position of the VD in the transverse plane was fixed, its longitudinal position was determined by looking at the residuals between VD and TPC. Once the VD was aligned, it could be considered as a stable reference system. The TPC and the ID were therefore aligned in the transverse plane with respect to the VD-OD system. The residuals

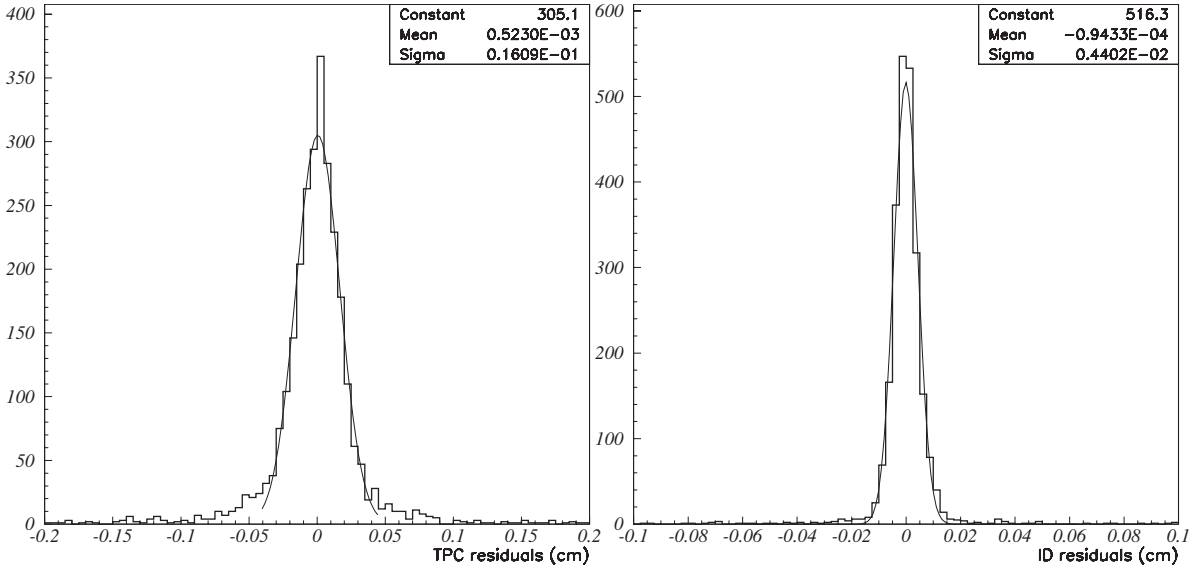

FIG. 23. Alignment of the DELPHI detector: $R\Phi$ residuals between VD+OD tracks (dimuons from Z^0 run in 1999) and hits in TPC (left) and ID (right). From Andreazza and Piotto, 1999.

after alignment with 1999 data are shown in Fig. 23.

The forward chambers were aligned relative to the VD, including its very forward part, the very forward tracker (VFT), and the ID. The TPC was not used because of distortion problems. FCB, the chamber further from the interaction point, was aligned first. Then FCA was aligned using the already aligned FCB in the track reconstruction. The last step was the alignment of the very forward tracker. Tracks fitted with VD, ID, FCA, and FCB were used plus the constraint from the beam spot. A consistent position of the VFT with respect to the other barrel detectors was enforced using the overlap between the first layer of pixels and the outer layer of the VD (see Fig. 24).

b. ALEPH

The ALEPH detector (Decamp *et al.*, 1990) was another experiment at the LEP collider. The alignment of its tracking devices is described by Wiedenmann (1992). The tracking detectors of ALEPH were the central TPC, a drift chamber called the inner tracking chamber (ITC), and a silicon VD. Alignment involved the determination of the global positions of TPC, ITC, and VD considering each component as a rigid body and the determination of TPC sectors and VD wafers relative to each other. The final global and local positions were obtained from real data from Z^0 decays, starting from survey measurements.

In the first step the ITC was aligned relative to the TPC. Three translations and three Euler angles were estimated by minimizing the residuals between the expected hit positions from the TPC and the observed hit positions in the ITC. In an analog way the VD was aligned relative to the ITC-TPC system.

The second step was the relative alignment of TPC sectors in the transverse plane using muons from Z^0 decays. Each muon track was fitted with three helix segments, one in the ITC, one in an inner sector, and one in an outer sector of the TPC. First the inner TPC sectors were aligned relative to the ITC helix segments, then the outer TPC sectors were aligned relative to the helix seg-

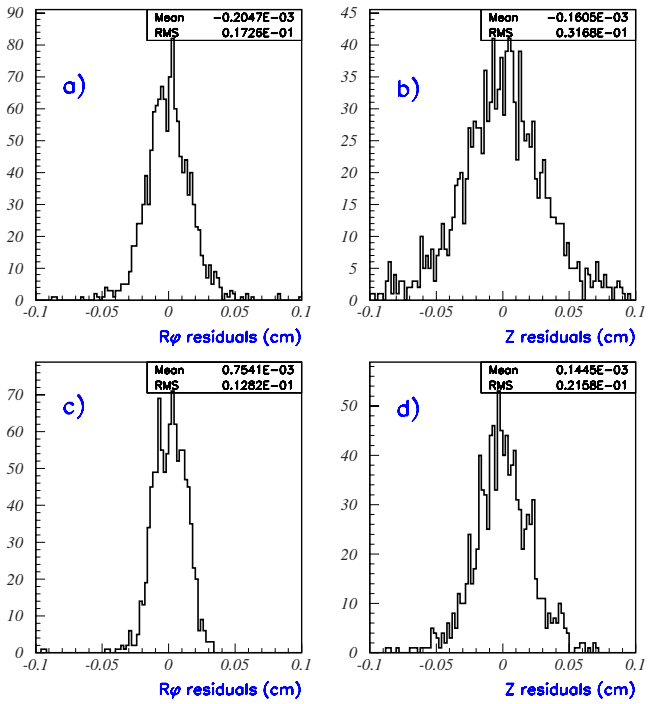

FIG. 24. (Color online) Alignment of the DELPHI detector: residuals in $R\Phi$ and z between hits in VFT and tracks fitted with VD, ID, FCA, and FCB. The upper plots are made for tracks in the region $15° \leqslant \theta \leqslant 21°$ (no overlap with VD) and lower plots in region $21° < \theta \leqslant 25°$ (overlap with VD). In both cases the mean values are at zero showing an alignment position consistent with both the barrel and the forward detectors. The sample of tracks is composed of muons from the Z^0 run in 1999. From Andreazza and Piotto, 1999.

ments in the ITC and the already aligned inner TPC sectors.

In the third step the VD was aligned internally using the concept of a constrained residual [for details see Wiedenmann (1992)]. Finally, TPC sectors were aligned longitudinally relative to the VD by fitting a single helix to the two tracks of a $Z^0 \to \mu^+\mu^-$ event, with the momentum constrained to the beam energy, and using only VD and ITC hits. For each TPC sector two alignment constants were determined, a longitudinal shift and a rotation around the sector center.

The entire procedure was iterated one more time using the previously determined alignment corrections. The final resolutions of the impact parameter of reconstructed tracks were 25 μm in the transverse and 29 μm in the longitudinal direction.

Although the alignment strategies of DELPHI and ALEPH were necessarily different in the details, some common features can be identified. In both cases a detector or pair of detectors served as a system of reference; in both cases the gaseous detectors were treated as rigid bodies; and in both cases the wafers of the silicon vertex detectors were aligned individually.

c. SLD

The vertex detector of the SLD experiment was of a size comparable to the ones of DELPHI and ALEPH, consisting of 96 alignable elements. The alignment procedure is described by Jackson *et al.* (2003, 2007). A novel feature is the solution of the alignment equations by singular value decomposition of the system matrix.

2. *B* factories

Track-based alignment in experiments at *B* factories (PEP-II at SLAC and KEKB at KEK) faces an environment similar to the *Z* factories, with the exception that track momenta are much lower.

a. BaBar

An example is the BaBar experiment at the PEP-II asymmetric collider at SLAC. The alignment strategy of the 340 wafers of its SVT is described in detail by Brown *et al.* (2007, 2009). The track sample used is a mix of physics triggers and cosmic rays passing near the nominal interaction point. Various cuts ensure high quality of tracks and an approximately uniform illumination of the SVT.

The selected tracks are fitted using the SVT and part of the surrounding drift chamber. For each wafer, a χ^2 is computed by summing over the residuals of all hits in the wafer and minimized independently for the alignment parameters of the wafer. The χ^2 can be extended to contain optical survey information. The minimization procedure is iterated until the parameters of all wafers stabilize. In the inner layers the curvature radius of the wafers is estimated in addition to the six degrees of freedom of translation and rotation.

The validation of the alignment procedure shows that global distortions can be reduced to a negligible level.

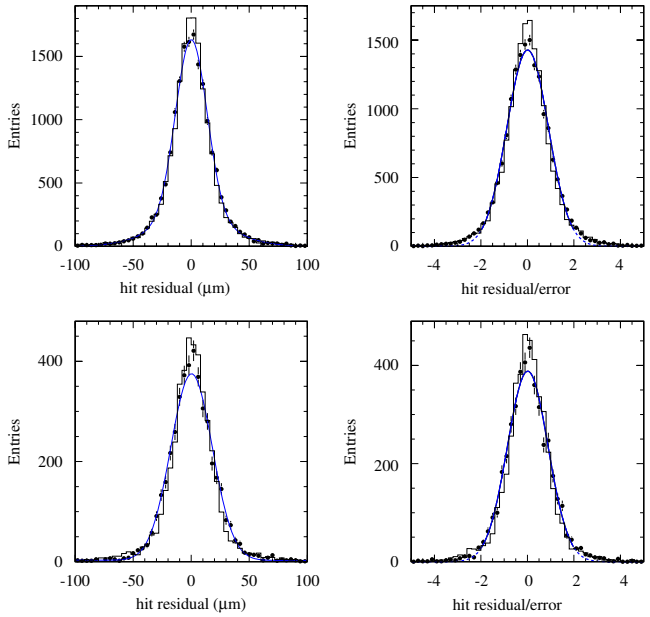

FIG. 25. (Color online) Single-hit residuals (left) and normalized residuals (right) from the inner three layers of the BaBar SVT. The top row corresponds to the u coordinate on each of the wafers and the bottom row to the v coordinate. The experimental data are shown as points and Monte Carlo simulation as histograms. The smooth curves are the results of a Gaussian fit to the data. From Brown *et al.*, 2009.

For instance, Fig. 25 shows hit residuals in the inner three layers of the SVT from selected $e^+e^- \to \mu^+\mu^-$ events. The top row shows residuals in the local u coordinate of the wafers, pointing in the direction of increasing Φ. The bottom row shows residuals in the local v coordinate of the wafers, orthogonal to u. The residual distributions are centered at zero, and the shapes of the data and Monte Carlo distributions are very similar. A Gaussian fit to the core of the distributions gives mean values consistent with zero and sigma values of 14 μm (13 μm) in data (Monte Carlo) for the local coordinate u and 18 μm (16 μm) data (Monte Carlo) for the local coordinate v.

b. BELLE

The silicon vertex detector of the BELLE experiment at KEK (Alimonti *et al.*, 2000) is smaller than the one of BaBar, comprising only 102 wafers. Its internal alignment using cosmic rays at zero magnetic field is described by Zhao (2000).

3. HERA

Concerning the complexity of the detectors, the experiments at the HERA electron-proton (ep) collider were closer to hadron collider experiments. The asymmetry of the beams was reflected in the marked asymmetry of the detectors.

a. H1

The H1 experiment was one of the two multipurpose detectors at the electron-proton collider HERA at DESY in Hamburg. It has a complex tracking system, consisting of a central, a forward, and a backward tracker (Abt *et al.*, 1997). The alignment strategy is described by Kleinwort (2007). The data sets used for track-based alignment include survey data, tracks from ep interactions, and cosmic muons. The central tracker is used as the frame of reference.

During the HERA-I data taking period (1992–2000) the central jet chamber (CJC) and the central silicon tracker were aligned and calibrated jointly using Millepede (Blobel and Kleinwort, 2002; Blobel, 2007). A total number of 465 alignment parameters and 774 calibration constants were estimated.

Several improvements were introduced for HERA-II data taking. Millepede was used for the online calibration of the time-dependent CJC parameters. The central silicon tracker was aligned internally using the singular value decomposition of the system matrix. The alignment relative to the CJC was achieved by treating it as a rigid body.

b. ZEUS

The other multipurpose detector at HERA was the ZEUS experiment. During HERA-II running it used a silicon microvertex detector (MVD), consisting of 600 barrel and 112 forward sensors. The alignment strategy is described by Mankel (2007). The track samples used included cosmic muons and tracks from ep collisions. Alignment with cosmic muons was done for individual ladders by an iterative χ^2 minimization algorithm (Kohno, 2006). Alignment with tracks from collisions was done for individual sensors, increasing the number of alignment parameters from 180 to 3560. Millepede was used for this alignment task.

4. Hadron colliders

Alignment of detectors at hadron colliders is faced with high data rates, a large background of low-energy tracks, and no well-defined center-of-mass energy. At LHC, the alignment problem is exacerbated by additional background because of pileup and in the case of ATLAS and CMS silicon trackers that are much larger than all of their predecessors. Both ATLAS and CMS have large muon detectors that need to be aligned relative to the central tracking system. As LHCb aims for extremely precise vertex reconstruction, it has to place strict constraints on its alignment.

a. DØ

The DØ experiment at the Fermilab Tevatron collider (Abazov *et al.*, 2006) has a silicon microstrip tracker (SMT) with 976 alignable elements, the same order of magnitude as the ZEUS MVD. The alignment procedure is described by Sopczak (2006, 2007). It is an iterative χ^2 minimization, local for each wafer. In the outer part of the central tracking system, the central fiber tracker can be aligned either simultaneously with the SMT or after the SMT has been aligned.

b. LHCb

The LHCb experiment (Augusto Alves, *et al.*, 2008) at the LHC is mainly dedicated to B physics. To achieve its physics goals it relies to a large extent on the performance of its precision VELO. The VELO (Behrendt, 2009) consists of 21 stations along the beam axis, each one divided into two modules. Each module supports two sensors, one measuring the radial coordinate (R) and the other one the azimuthal coordinate (Φ). While it is not a large system, it has particularly demanding alignment requirements because the modules, each with a single hit resolution below 10 μm in both coordinates, have to be retracted by 3 cm from the beam line while the machine is filled. As the alignment of the VELO has a significant effect on the high-level trigger, a fast but nevertheless precise alignment should be possible at any time (Viret, 2007).

The track-based alignment of the VELO is described by Viret (2007), Viret *et al.* (2008), and Gersabeck (2009). The track sample used is a mixture of tracks from the primary vertex and tracks from beam halo or beam-gas interactions. The first step of the alignment is the relative alignment of the two sensors on each module. The second step is the relative alignment of the modules in each half of the VELO. Both steps use a global minimization method based on the Millepede algorithm. The third step is the alignment of the two halves with respect to each other. This requires tracks that pass through both halves. As only six relative alignment constants have to be determined, global minimization using matrix inversion is used. The performance of the alignment algorithm has been assessed with test beam data (Gersabeck *et al.*, 2008). The accuracy obtained is about 2 μm for translations and 0.1 mrad for rotations. The global alignment strategy of LHCb is described by Baldini *et al.* (2006, 2007).

c. ALICE

The ALICE experiment (Aamodt *et al.*, 2008) at the LHC is dedicated to heavy-ion physics. Its inner tracking system (ITS) (ALICE Collaboration, 1999) is composed of 2198 silicon pixel, silicon drift, and silicon strip modules, which is modest compared to ATLAS or CMS but one order of magnitude higher than in the heavy-ion experiment STAR (Margetis *et al.*, 2007; Fisyak *et al.*, 2008). The ITS has to be aligned internally as well as relative to the TPC. Both a local iterative method based on the minimization of residuals and a global method based on Millepede are envisaged for this task. The muon spectrometer of ALICE will be aligned using Millepede (Castillo, 2007).

d. ATLAS

The ATLAS experiment (Aad *et al.*, 2008) at the LHC has an inner detector (ATLAS Collaboration, 1997a,

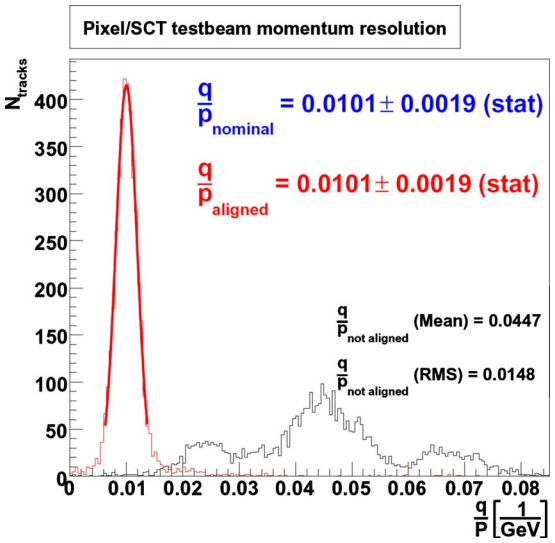

FIG. 26. (Color online) Alignment of the ATLAS inner detector: momentum resolution of pixel and SCT modules for a combined test beam run of 100 GeV π^+ without alignment corrections (nonfitted histogram) and with alignment corrections derived with the local χ^2 method (Gaussian-fitted histogram). From Gonzalez-Sevilla, 2008.

1997b) that consists of two parts, a silicon tracking system closer to the interaction point and a straw-tube tracking system, the transition radiation tracker (TRT). The silicon tracker system consists of 5832 individual silicon modules, arranged in three pixel (PIX) layers and four strip (SCT) layers in the barrel and three PIX and nine SCT disks in each of the end caps. Track-based alignment is described by Escobar (2006), Brückman de Renstrom (2007a, 2007b), Karagöz Ünel (2007), and Morley (2008). A number of independent alignment algorithms have been implemented and tested. The first is a global χ^2 minimization. Various methods for solving the resulting large system have been tested. The second one is an iterated local χ^2 minimization. The third one is called robust alignment and is suitable to align any kind of detector with overlapping sensors. Alignment of the TRT can be done by local χ^2 minimization using tracks extrapolated from the silicon tracking system (PIX +SCT) or internally using global χ^2 minimization (Bocci and Hulsbergen, 1997). The track-based alignment of the ATLAS muon spectrometer is described by Kortner *et al.* (2007).

The alignment procedures have been studied on simulated data and validated with test beam and cosmics data. As an example, Fig. 26 shows the momentum of PIX and SCT modules in the test beam before and after alignment with the local χ^2 method.

e. CMS

The all-silicon tracker of the CMS experiment (Chatrchyan *et al.*, 2008) at the LHC has an area of about 200 m^2 and thus is the largest that has ever been built (CMS Collaboration, 1998, 2000). The pixel part consists of 1440 alignable modules, of which 768 are in the barrel and 672 in the end caps; the silicon microstrip part consists of 11 860 alignable modules, of which 6084 are in the barrel and 5776 in the end caps. Summaries of the alignment strategy are given by Ronga (2007) and Weber (2007). In addition to collision data such as $Z^0 \to \mu^+\mu^-$ events, cosmic muons and beam halo muons will be used. In addition, vertex and mass constraints can be used to better constrain the alignment parameters and to suppress weak modes.

Three algorithms have been implemented to solve the CMS alignment challenge. The first one is global optimization with Millepede using the GMRES method for solving the large system of linear equations (Schleper *et al.*, 2008). The second one is global optimization by the Kalman alignment algorithm (KAA) (Widl *et al.*, 2006; Frühwirth and Widl, 2007; Widl and Frühwirth, 2008). The third one is local optimization by the hit and impact point (HIP) algorithm (Karimäki *et al.*, 2003, 2006).

All algorithms have been studied individually on simulated data. Results are reported by Karimäki *et al.* (2006), Schleper *et al.* (2008), and Widl and Frühwirth (2009). For example, the entire tracker with 44 432 alignment parameters was aligned with Millepede using 2×10^6 μ pairs and imposing a constraint on the invariant mass (Schleper *et al.*, 2008). The final alignment precisions of the most sensitive coordinate were about 1.2 μm for the pixel barrel modules, 2.5 μm for the pixel end-cap modules, about 10 μm for the strip barrel modules, and about 23 μm for the strip end-cap modules.

The alignment of large tracking detectors such as the CMS tracker can be speeded up considerably by parallelization. The Millepede alignment of the tracker, performed for the 2008 Computing, Software and Analysis Challenge (CSA08), took about 1.5 h on 50 cores for the preparation of the large matrix and 5 h on a single CPU for solving the linear system and the computation of the final result (CMS Collaboration, 2009a). The alignment with the KAA took about 15 h on ten cores (Widl and Frühwirth, 2009).

A comparison of the three methods with cosmics data, recorded at the tracker integration facility of CMS, is reported by Adam *et al.* (2009a, 2009b). About 2000 modules, 15% of the entire tracker, were activated. More than 4×10^6 cosmic triggers were recorded at different temperatures and different trigger conditions. The comparison shows that in the barrel HIP and KAA performed equally well, Millepede being slightly worse. In the end caps all methods performed on the same level. As an example, Fig. 27 shows the hit residuals of the three methods in the barrel, separately for single- and double-sided modules in the inner and outer barrels, respectively.

V. CONCLUSIONS

This review has shown that adaptive methods of track and vertex reconstruction are in widespread use today. They have been adopted by the large LHC experiments, but it seems that their impact has been greater in vertex

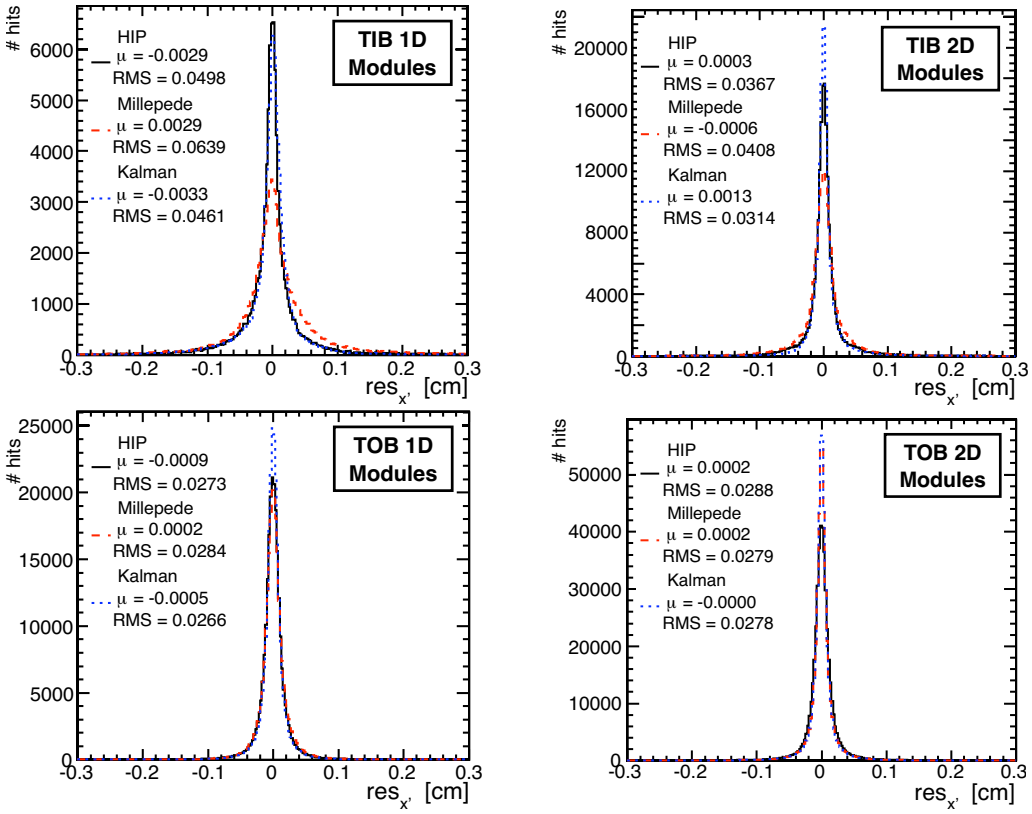

FIG. 27. (Color online) Alignment of the CMS tracker: hit residuals from three track-based algorithms: HIP (solid), Millepede (dashed), and Kalman alignment algorithm (dotted). Two tracker subdetectors are shown: tracker inner barrel (TIB) in the top row and tracker outer barrel (TOB) in the bottom row. The residuals are in the local x direction perpendicular to the strips. Top left: TIB, single-sided modules; top right: TIB, double-sided modules; bottom left: TOB, single-sided modules; and bottom right: TOB, double-sided modules. The track fit is restricted to modules aligned by all three algorithms. Adapted from Adam *et al.*, 2009a.

reconstruction than in track reconstruction. We can think of two potential reasons for that. First, in spite of the large track multiplicity, track finding and fitting can still be comfortably accomplished by traditional or hybrid methods, such as the combinatorial Kalman filter. Second, the reconstruction of the decay vertices of very short-lived particles such as B mesons is sufficiently difficult to give adaptive methods an edge over the traditional ones. In any case it can be stated that the traditional distinction between pattern recognition, on the one hand, and estimation and testing, on the other hand, has begun to dissolve and has been superseded by dynamic competition between different hypotheses about what constitutes a track or a vertex.

Not all methods discussed here exhibit the same degree of adaptivity. For example, the deterministic annealing filter is "more adaptive" than the Gaussian-sum filter (GSF), as it does not require an explicit outlier model and thus manages with less assumptions about the data. The GSF in turn is more adaptive than a plain Kalman filter, as it explores several hypotheses about the observations or the process noise in parallel until in the end one of them is selected as the winner or the most probable ones are combined to the final result.

The first method that merits the label adaptive was the application of the Hopfield network to track finding. The structure of the network is not fixed but determined by the data, and each state of the network represents a hypothesis about which hit belongs to which track. In this case the competition takes place sequentially as each state of the network is superseded by a better one due to the dynamics of the update. The incorporation of a physical track model into the Hopfield network has never been tried, and indeed it is difficult to see how this could be accomplished. For this reason the Hopfield network is not able to deliver a statistically optimal estimate of the track parameters. The concept of the elastic arm or deformable template, namely, the merging of a continuous estimation problem with a combinatorial optimization problem, was able to overcome this limitation but—in the original formulation—at the price of a nontrivial minimization task. The introduction of a temperature parameter gave the user the choice either to enforce a hard decision about which hit belongs to which template or to retain a probabilistic element by not cooling down to zero. After it had been recognized that the EM algorithm was an alternative method of minimizing the error function or energy function of the deformable template, the entire concept could be transferred to the well-known realm of least-squares estimation and stochastic filters. It turned out that, with the Kalman filter as the basic building block, the construction of an adaptive filter actually required little additional work. With equally small effort other least-squares estimators can

be upgraded to an adaptive version as well. As vertex fitting is much more independent of the experimental setup than track fitting, adaptive vertex reconstruction algorithms could be made available in a toolbox that can easily be interfaced with experiment specific software.

In the field of alignment algorithms there is a promising candidate for a general purpose, experiment independent method: the Millepede algorithm. Nevertheless the two largest experiments at LHC have implemented one or even several alternative solutions. This is no doubt a wise decision as the results of even a fairly mature method such as Millepede will be accepted with more confidence if they are cross validated our independent algorithms.

VI. OUTLOOK

In spite of the apparent success of adaptive methods there remain a number of open questions and research problems. In the following we give a list of problems to be tackled, with no claim of being exhaustive. Some of them might not yet be urgent but might get so at an upgraded hadron collider, for instance, the super-LHC (SLHC) (Scandale and Zimmermann, 2007). The reader should be aware that the following items are to some extent speculative and inevitably tinted by our experiences, predilections, and expectations.

A. Parallelization of track finding and fitting

The move to multicore technology in computing has opened up possibilities that could lead to substantial gains in the speed of track reconstruction. Consequently, methods that exploit the parallelism in the hardware, both multicore and data level parallelism [single instruction multiple data (SIMD)], would be a major step ahead in the reconstruction code. For instance, the SIMDized Kalman-filter based track fit described by Gorbunov *et al.* (2008) takes only about 1 μs per track, which is 10 000 times faster than the original version. Going beyond the simple Kalman filter, parallelization of the combinatorial Kalman filter would be a promising research project, with a potentially large impact on the methodology of track finding. This is connected with the problem how to use tracker information in the first level trigger at the SLHC. Full track finding in the first level seems out of the question today, but further research in this direction may overcome this limitation.

B. Track finding or fitting at very high noise levels

At the SLHC the track multiplicity and the noise level will be even higher than today at the LHC. In this case an attractive alternative to the combinatorial Kalman filter could be a fast global pattern recognition, e.g., based on methods from signal processing and image analysis, and a subsequent adaptive filter to sort out the details of the hit-track association.

C. Track reconstruction in narrow jets

In narrow jets the hit-track association could benefit from a multitrack filter, characterized by a competition between all tracks for all hits in any given detector element crossed by the jet. The main problem to be solved is the initialization of the filter with the correct number of templates. Also, it has not been conclusively shown yet that the simultaneous competition of all tracks in the jet for the hits is actually superior to a sequential application of single adaptive filters.

D. Adaptive vertex reconstruction

A similar question can be posed in the task of vertex reconstruction: Is the multivertex fit really superior to a sequential application of the adaptive vertex fit, and if so in which circumstances? A study with decay vertices of very short-lived particles should be able to shed some light on this question.

E. Non-Gaussian phenomena

One of the available solutions to non-Gaussian effects such as bremsstrahlung is the Gaussian-sum filter, which approximates the non-Gaussian density in question by a mixture of Gaussians. This is not entirely satisfactory because of two reasons: it is only an approximation and it is computationally expensive. It would be interesting to study whether a filter using the actual densities could be formulated and made computationally feasible. It still would have to involve approximations as the propagation and the update of the track state would have to be computed by numerical convolution and numerical integration. On the other hand, the exponential rise in the number of components and time-consuming collapsing procedures could be avoided.

F. Usage of posterior weights

Some adaptive methods, because of their Bayesian flavor, return several hypotheses about a track or vertex, each one with its posterior weight. The question arises how to use this information in subsequent stages of data analysis. One example is the fit of an electron track with the Gaussian-sum filter, the final state being a mixture of Gaussians. If the electron track is to be used in a vertex fit, there are two possibilities: the mixture is collapsed to a single Gaussian and processed by an adaptive or least-squares fit or the entire mixture state is processed by a Gaussian-sum vertex filter. The latter approach can be expected to give the better results, but this has yet to be shown conclusively. A more difficult question arises if kinematic constraints are to be imposed on the results of an adaptive vertex fit: How are the assignment probabilities of the tracks to the vertex to be treated by the kinematic fit? These and similar questions need to be studied if the full potential of adaptive methods is to be exploited.

G. Estimation of material

A general problem, which to our knowledge has so far been not met with a general solution, is the estimation of the amount and the distribution of the material in a tracking detector from reconstructed tracks. In the track simulation program usually a detailed detector description is available so that there is a close correspondence between the detector model and the actual detector. In track reconstruction a simplified detector model is employed in most cases mainly because of the amount of computing time that is considered tolerable. The simplified model often reflects only inadequately the actual amount and location of the material and often assumes that the material is concentrated in thin layers when it is actually not. The consequences are unsatisfactory χ^2 distributions and biased test statistics. The solution of this inverse problem is in our opinion a challenging but interesting and important research topic.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for various corrections and numerous suggestions that have helped to improve the paper substantially.

REFERENCES

Aad, G., *et al.* (ATLAS Collaboration), 2008, J. Instrum. **3**, S08003.
Aamodt, K., *et al.* (ALICE Collaboration), 2008, J. Instrum. **3**, S08002.
Abazov, V. M., *et al.* (DØ Collaboration), 2006, Nucl. Instrum. Methods Phys. Res. A **565**, 463.
Abreu, P., *et al.* (DELPHI Collaboration), 1996, Nucl. Instrum. Methods Phys. Res. A **378**, 57.
Abt, I., *et al.* (H1 Collaboration), 1997, Nucl. Instrum. Methods Phys. Res. A **386**, 348.
Adam, W., *et al.*, 2005, J. Phys. G **31**, N9.
Adam, W., *et al.*, 2009a, J. Instrum. **4**, T07001.
Adam, W., *et al.*, 2009b, Technical Report No. CMS NOTE 2009/002, CERN.
Agakichiev, G., *et al.*, 1997, Nucl. Instrum. Methods Phys. Res. A **394**, 225.
Akimoto, T., *et al.*, 2006, Nucl. Instrum. Methods Phys. Res. A **556**, 459.
Aktas, A., *et al.* (H1 Collaboration), 2005, Eur. Phys. J. C **40**, 349.
Aktas, A., *et al.* (H1 Collaboration), 2006, Eur. Phys. J. C **47**, 597.
Alexopoulos, T., *et al.*, 2008, Nucl. Instrum. Methods Phys. Res. A **592**, 456.
ALICE Collaboration, 1999, Technical Report No. CERN/LHCC 99-12, CERN.
Alimonti, G., *et al.*, 2000, Nucl. Instrum. Methods Phys. Res. A **453**, 71.
Amsler, C., *et al.* (Particle Data Group), 2008, Phys. Lett. B **667**, 1.
Andreazza, A., and E. Piotto, 1999, Technical Report No. DELPHI 99-153, CERN.
ATLAS Collaboration, 1997a, Technical Report No. CERN/LHCC 97-16, CERN, Vol. I-II.
ATLAS Collaboration, 1997b, Technical Report No. CERN/LHCC 97-17, CERN, Vol. I-II.
Augusto Alves Jr., A., *et al.* (LHCb Collaboration), 2008, J. Instrum. **3**, S08005.
Badalà, A., *et al.*, 2003, Nucl. Instrum. Methods Phys. Res. A **502**, 503.
Badalà, A., *et al.*, 2004, Nucl. Instrum. Methods Phys. Res. A **534**, 211.
Baginyan, S., *et al.*, 1994, Comput. Phys. Commun. **79**, 165.
Baldini, W., *et al.*, 2006, Technical Report No. CERN-LHCb-2006-035, CERN.
Baldini, W., *et al.* (for the LHCb Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 179–222.
Barbone, L., *et al.*, 2006, Nucl. Instrum. Methods Phys. Res. A **566**, 45.
Bates, D. M., and D. G. Watts, 1988, *Nonlinear Regression Analysis and Its Applications* (Wiley, New York).
Behrendt, O. (on behalf of the LHCb VELO Group), 2009, Nucl. Instrum. Methods Phys. Res. A **598**, 61.
Bethe, H., and W. Heitler, 1934, Proc. R. Soc. London, Ser. A **146**, 83.
Billoir, P., 1984, Nucl. Instrum. Methods Phys. Res. **225**, 352.
Billoir, P., 1989, Comput. Phys. Commun. **57**, 390.
Billoir, P., R. Frühwirth, and M. Regler, 1985, Nucl. Instrum. Methods Phys. Res. A **241**, 115.
Billoir, P., and S. Qian, 1990, Nucl. Instrum. Methods Phys. Res. A **295**, 492.
Billoir, P., and S. Qian, 1992, Nucl. Instrum. Methods Phys. Res. A **311**, 139.
Billoir, P., and S. Qian, 1994, Nucl. Instrum. Methods Phys. Res. A **350**, 624.
Blankenbecler, R., 1994, Comput. Phys. Commun. **81**, 318.
Blobel, V., 2006, Nucl. Instrum. Methods Phys. Res. A **566**, 5.
Blobel, V., 2007, Technical Report, Universität Hamburg.
Blobel, V., and C. Kleinwort, 2002, *Proceedings of the 2002 Conference on Advanced Statistical Techniques in Particle Physics* (University of Durham, Durham, UK).
Blusk, S., *et al.*, 2007, Ed., in Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004.
Bocci, A., and W. Hulsbergen, 1997, Technical Report No. ATL-INDET-PUB-2007-009, CERN.
Bock, R. K., and A. Vasilescu, 1998, *The Particle Detector Brief Book* (Springer, New York).
Brown, D., A. Gritsan, and D. Roberts, 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 29–39.
Brown, D. N., *et al.*, 2009, Nucl. Instrum. Methods Phys. Res. A **603**, 467.
Brückman de Renstrom, P., 2004, Technical Report No. DELPHI 2004-047, CERN.
Brückman de Renstrom, P. (on behalf of the ATLAS Inner Detector Alignment Community), 2007a, Nucl. Instrum. Methods Phys. Res. A **582**, 184.
Brückman de Renstrom, P. (for the ATLAS Collaboration), 2007b, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 147–158.
Bugge, L., and J. Myrheim, 1981, Nucl. Instrum. Methods **179**, 365.
Castillo, J. (for the ALICE Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical

Report No. CERN-2007-004, pp. 127–138.
Catlin, D. E., 1989, *Estimation, Control, and the Discrete Kalman Filter* (Springer, New York).
CERN Indico, 2007, Second LHC Detector Alignment Workshop (unpublished).
CERN Indico, 2009, Third LHC Detector Alignment Workshop (unpublished).
Chabanat, E., *et al.*, 2005, Nucl. Instrum. Methods Phys. Res. A **549**, 188.
Chatrchyan, S., *et al.* (CMS Collaboration), 2008, J. Instrum. **3**, S08004.
Chernov, N., *et al.*, 1993, Comput. Phys. Commun. **74**, 217.
CMS Collaboration, 1998, Technical Report No. CERN/LHCC 98-6, CERN.
CMS Collaboration, 2000, Technical Report No. CERN/LHCC 2000-016, CERN.
CMS Collaboration, 2009a, private communication.
CMS Collaboration, 2009b, Technical Report No. CMS IN-2008/044, CERN.
Costa, M. (on behalf of the ATLAS Inner Detector Group), 2007, Nucl. Instrum. Methods Phys. Res. A **582**, 785.
Cucciarelli, S., 2005, Nucl. Instrum. Methods Phys. Res. A **549**, 49.
Das, S. R., 1973, IEEE Trans. Comput. **C-22**, 187.
Decamp, D., *et al.* (ALEPH Collaboration), 1990, Nucl. Instrum. Methods Phys. Res. A **294**, 121.
Dempster, A. P., N. M. Laird, and D. B. Rubin, 1977, J. R. Stat. Soc. Ser. B (Methodol.) **39**, 1.
Denby, B., 1988, Comput. Phys. Commun. **49**, 429.
Denby, B., 1999, Comput. Phys. Commun. **119**, 219.
D'Hondt, J., *et al.*, 2004, IEEE Trans. Nucl. Sci. **51**, 2037.
Diehl, M., *et al.*, 1997, Nucl. Instrum. Methods Phys. Res. A **389**, 180.
Dror, G., and E. Etzion, 2000, *Proceedings of the VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research* (Fermi National Accelerator Laboratory, Batavia, IL).
Durbin, R., and D. Willshaw, 1987, Nature (London) **326**, 689.
Dydak, F., 2003, Technical Report No. HARP 03-001, CERN.
Eichinger, H., and M. Regler, 1981, Technical Report No. CERN 81-06, CERN.
Erdmann, W., 2006, Nucl. Instrum. Methods Phys. Res. A **560**, 89.
Escobar, C. (on behalf of the ATLAS Inner Detector Alignment Group), 2006, Conference Record of 2006 IEEE Nuclear Science Symposium San Diego (unpublished), Vol. 3, p. 1643.
Fisyak, Y. V., *et al.*, 2008, J. Phys.: Conf. Ser. **119**, 032017.
Frühwirth, R., 1987, Nucl. Instrum. Methods Phys. Res. A **262**, 444.
Frühwirth, R., 1993, Comput. Phys. Commun. **78**, 23.
Frühwirth, R., 1995, Comput. Phys. Commun. **85**, 189.
Frühwirth, R., 1997, Comput. Phys. Commun. **100**, 1.
Frühwirth, R., 2003, Comput. Phys. Commun. **154**, 131.
Frühwirth, R., and S. Frühwirth-Schnatter, 1998, Comput. Phys. Commun. **110**, 80.
Frühwirth, R., and T. Speer, 2004, Nucl. Instrum. Methods Phys. Res. A **534**, 217.
Frühwirth, R., and A. Strandlie, 1999, Comput. Phys. Commun. **120**, 197.
Frühwirth, R., and A. Strandlie, 2006, Nucl. Instrum. Methods Phys. Res. A **559**, 162.
Frühwirth, R., T. Todorov, and M. Winkler, 2003, J. Phys. G **29**, 561.
Frühwirth, R., and W. Waltenberger, 2004, Technical Report No. CMS CR 2004/062, CERN.
Frühwirth, R., and W. Waltenberger, 2008, Austrian J. Statist. **37**, 301.
Frühwirth, R., and E. Widl, 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 13–19.
Frühwirth, R., *et al.*, 1996, Comput. Phys. Commun. **96**, 189.
Frühwirth, R., *et al.*, 2000, *Data Analysis Techniques for High-Energy Physics*, 2nd ed. (Cambridge University Press, Cambridge).
Frühwirth, R., *et al.*, 2003a, Nucl. Instrum. Methods Phys. Res. A **502**, 699.
Frühwirth, R., *et al.*, 2003b, Proceedings of the 2003 Conference on Computing in High Energy and Nuclear Physics (SLAC) (unpublished).
Gersabeck, M., 2009, Nucl. Instrum. Methods Phys. Res. A **598**, 71.
Gersabeck, M., *et al.*, 2008, Nucl. Instrum. Methods Phys. Res. A **596**, 164.
Glazov, A., *et al.*, 1993, Nucl. Instrum. Methods Phys. Res. A **329**, 262.
Golling, T. (for the ALICE, ATLAS, CMS, and LHCb Collaborations), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 95–104.
Golutvin, I. A., *et al.*, 2000, Comput. Phys. Commun. **126**, 72.
Gonzalez-Sevilla, S. (on behalf of the ATLAS Collaboration), 2008, J. Phys.: Conf. Ser. **119**, 032019.
Gorbunov, S., and I. Kisel, 2005, Technical Report No. CBM-SOFT-note-2005-002, GSI.
Gorbunov, S., and I. Kisel, 2006, Nucl. Instrum. Methods Phys. Res. A **559**, 139.
Gorbunov, S., *et al.*, 2008, Comput. Phys. Commun. **178**, 374.
Grote, H., 1987, Rep. Prog. Phys. **50**, 473.
Guttman, I., and D. Peña, 1985, Technical Report No. 1985/1, University of Toronto.
Gyulassy, M., and M. Harlander, 1991, Comput. Phys. Commun. **66**, 31.
Gyulassy, M., and M. Harlander, 1992, Nucl. Instrum. Methods Phys. Res. A **316**, 238.
Haas, A., 2000, Proceedings of the VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (FNAL) (unpublished).
Hampel, F. R., *et al.*, 1986, *Robust Statistics: The Approach Based on the Influence Function* (Wiley, New York).
Hansroul, M., H. Jeremie, and D. Savard, 1988, Nucl. Instrum. Methods Phys. Res. A **270**, 498.
Harr, R., 1995, IEEE Trans. Nucl. Sci. **42**, 134.
Hertz, J., A. Krogh, and R. G. Palmer, 1991, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood).
Hillert, S., 2008, Technical Report No. LC-DET-2008-004, DESY.
Hopfield, J. J., 1982, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554.
Hopfield, J. J., and D. W. Tank, 1985, Biol. Cybern. **52**, 141.
Hough, P. V. C., 1959, *Proceedings of the International Conference on High Energy Accelerators and Instrumentation* (CERN, Geneva).
Huber, P. J., 1981, *Robust Statistics* (Wiley, New York).
Jackson, D. J., 1997, Nucl. Instrum. Methods Phys. Res. A **388**, 247.

Jackson, D. J., D. Su, and F. J. Wickens, 2003, Nucl. Instrum. Methods Phys. Res. A **510**, 233.
Jackson, D. J., D. Su, and F. J. Wickens (for the SLD Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 59–69.
James, F., 1983, Nucl. Instrum. Methods Phys. Res. **211**, 145.
Jobes, M., and H. R. Shaylor, 1972, Rep. Prog. Phys. **35**, 1077.
Karagöz Ünel, M. (representing the ATLAS Inner Detector Alignment Group), 2007, Nucl. Phys. B (Proc. Suppl.) **172**, 194.
Karimäki, V., 1997, Technical Report No. CMS NOTE 1997/051, CERN.
Karimäki, V., T. Lampen, and F.-P. Schilling, 2006, Technical Report No. CMS NOTE 2006/018, CERN.
Karimäki, V., *et al.*, 2003, Technical Report No. CMS CR 2003/022, CERN.
Kartvelishvili, V. (on behalf of the ATLAS Collaboration), 2007, Nucl. Phys. B (Proc. Suppl.) **172**, 208.
Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, 1983, Science **220**, 671.
Kisel, I., 2006, Nucl. Instrum. Methods Phys. Res. A **566**, 85.
Kisel, I., E. Konotopskaya, and V. Kovalenko, 1997, Nucl. Instrum. Methods Phys. Res. A **389**, 167.
Kisel, I., *et al.*, 1997, Nucl. Instrum. Methods Phys. Res. A **387**, 433.
Kisel, I. V., V. N. Neskoromnyĭ, and G. A. Ososkov, 1993, Phys. Part. Nucl. **24**, 657.
Kitagawa, G., 1989, Comput. Math. Appl. **18**, 503.
Kitagawa, G., 1994, Ann. Inst. Stat. Math. **46**, 605.
Kleinwort, C. (for the H1 Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 41–49.
Kohno, T., 2006, Nucl. Instrum. Methods Phys. Res. A **559**, 153.
Kortner, O., *et al.*, 2007, Nucl. Instrum. Methods Phys. Res. A **581**, 545.
Kruskal, J. B., 1956, Proc. Am. Math. Soc. **7**, 48.
Lampén, T., *et al.*, 2008, Technical Report No. CMS NOTE-2008/029, CERN.
Laurikainen, P., W. G. Moorhead, and W. Matt, 1972, Nucl. Instrum. Methods **98**, 349.
Leo, W. R., 1994, *Techniques for Nuclear and Particle Physics Experiments: A How to Approach*, 2nd ed. (Springer, New York).
Lin, D., and I. Guttman, 1993, Stat. Probab. Lett. **16**, 259.
Lindsey, C. S., and B. Denby, 1991, Nucl. Instrum. Methods Phys. Res. A **302**, 217.
Lindström, M., 1995, Nucl. Instrum. Methods Phys. Res. A **357**, 129.
Mankel, R., 1997, Nucl. Instrum. Methods Phys. Res. A **395**, 169.
Mankel, R., 2004, Rep. Prog. Phys. **67**, 553.
Mankel, R. (for the ZEUS Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 51–58.
Margetis, S., *et al.* (for the STAR Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 23–28.
Morley, A. (on behalf of the ATLAS Inner Detector Alignment Group), 2008, Nucl. Instrum. Methods Phys. Res. A **596**, 32.
Ohlsson, M., 1993, Comput. Phys. Commun. **77**, 19.
Ohlsson, M., C. Peterson, and A. L. Yuille, 1992, Comput. Phys. Commun. **71**, 77.
Peterson, C., 1989, Nucl. Instrum. Methods Phys. Res. A **279**, 537.
Peterson, C., and J. R. Anderson, 1987, Complex Syst. **1**, 995.
Piacquadio, G., K. Prokofiev, and A. Wildauer, 2008, J. Phys.: Conf. Ser. **119**, 032033.
Regler, M., R. Frühwirth, and W. Mitaroff, 1996, Int. J. Mod. Phys. C **7**, 521.
Ronga, F. J. (for the CMS Collaboration), 2007, Nucl. Phys. B (Proc. Suppl.) **172**, 202.
Rousseeuw, P. J., and A. M. Leroy, 1987, *Robust Regression and Outlier Detection* (Wiley, New York).
Scandale, W., and F. Zimmermann, 2007, CERN Cour. **47**, 32.
Schleper, P., G. Steinbrück, and M. Stoye, 2008, J. Phys.: Conf. Ser. **119**, 032040.
Sopczak, A. (on behalf of the DØ Collaboration), 2006, Nucl. Instrum. Methods Phys. Res. A **566**, 142.
Sopczak, A. (for the DØ Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 71–79.
Speer, T., and R. Frühwirth, 2006, Comput. Phys. Commun. **174**, 935.
Speer, T., *et al.*, 2006a, Nucl. Instrum. Methods Phys. Res. A **566**, 149.
Speer, T., *et al.*, 2006b, Nucl. Instrum. Methods Phys. Res. A **559**, 143.
Speer, T., *et al.*, 2006c, Technical Report No. CMS NOTE 2006/032, CERN.
Stepanov, N., and A. Khanov, 1997, Nucl. Instrum. Methods Phys. Res. A **389**, 177.
Stimpfl-Abele, G., and L. Garrido, 1991, Comput. Phys. Commun. **64**, 46.
Strandlie, A., 2000, Ph.D. thesis (University of Oslo).
Strandlie, A., and R. Frühwirth, 2000, Comput. Phys. Commun. **133**, 34.
Strandlie, A., and R. Frühwirth, 2006a, IEEE Trans. Nucl. Sci. **53**, 3842.
Strandlie, A., and R. Frühwirth, 2006b, Nucl. Instrum. Methods Phys. Res. A **566**, 157.
Strandlie, A., and W. Wittek, 2006, Nucl. Instrum. Methods Phys. Res. A **566**, 687.
Strandlie, A., J. Wroldsen, R. Frühwirth, and B. Lillekjendlie, 2000, Comput. Phys. Commun. **131**, 95.
Strandlie, A., and J. Zerubia, 1999, Comput. Phys. Commun. **123**, 77.
Vanlaer, P., *et al.*, 2006, Technical Report No. CMS NOTE 2006/029, CERN.
Viret, S. (on behalf of the LHCb VELO Group), 2007, Nucl. Instrum. Methods Phys. Res. A **582**, 806.
Viret, S., C. Parkes, and M. Gersabeck, 2008, Nucl. Instrum. Methods Phys. Res. A **596**, 157.
Viret, S., C. Parles, and D. Petrie, 2005, Technical Report No. LHCb-2005-101, CERN.
Waltenberger, W., 2004, Ph.D. thesis (University of Technology, Vienna).
Waltenberger, W., 2008, Technical Report No. CMS NOTE-2008/033, CERN.
Waltenberger, W., R. Frühwirth, and P. Vanlaer, 2007, J. Phys. G **34**, N343.
Waltenberger, W., W. Mitaroff, and F. Moser, 2007, Nucl. Instrum. Methods Phys. Res. A **581**, 549.
Waltenberger, W., and F. Moser, 2006, Proceedings of the

IEEE 2006 IEEE Nuclear Science Symposium Conference Record (unpublished), Vol. 1, p. 104.
Weber, M. (for the CMS Collaboration), 2007, Proceedings of the First LHC Detector Alignment Workshop, Technical Report No. CERN-2007-004, pp. 95–104.
Wicke, D., 1998, Technical Report No. DELPHI 98-163, CERN.
Widl, E., and R. Frühwirth, 2008, J. Phys.: Conf. Ser. **119**, 032038.
Widl, E., and R. Frühwirth, 2009, Technical Report No. CMS CR-2009/171, CERN.
Widl, E., R. Frühwirth, and W. Adam, 2006, Technical Report No. CMS NOTE 2006/022, CERN.
Wiedenmann, W. (ALEPH Collaboration), 1992, Technical Report No. CERN-PPE/92-90, CERN.
Wildauer, A., 2006, Ph.D. thesis (Leopold-Franzens-Universität Innsbruck).
Winkler, M., 2002, Ph.D. thesis (University of Technology, Vienna).
Yuille, A., 1990, Neural Comput. **2**, 1.
Yuille, A., P. Stolorz, and J. Utans, 1994, Neural Comput. **6**, 334.
Zhao, H. W., 2000, Technical Report No. Belle-Note 348, KEK.