

Statistics for trajectometry

Pierre Billoir

¹LPNHE, Paris (IN2P3)

Abstract. A trajectometer is made of layers of charged particle detectors which measure successive positions along the trajectories; it is generally immersed in a magnetic field, so the curvature of the trajectory provides a measurement of the momentum. A method to perform a progressive fitting of the trajectory (Kalman Filter formalism), incorporating the measurements one after one, with an optimal account for the perturbations (multiple scattering, energy loss), is described with some indications for practical implementations in realistic detector layouts. Useful byproducts of the method and tests of validity are discussed. The procedure appears to be a combination *ad libitum* of elementary operations on vectors and matrices of fixed dimension (the number of parameters needed to define the trajectory), affording very flexible strategies, including a coupling of the pattern recognition of tracks with the fit of the trajectory, and combination with calorimetric or timing measurements. Extension to non-gaussian errors is discussed.

Once the trajectories of an event are independently reconstructed, they may be extrapolated back to the region of production of the particles (target, or zone of intersection of the beams in a collider) and associated to one or several vertices (primary interaction, and possible secondary interactions or decays): a fast and flexible method is described to perform these operations and improve the geometrical reconstruction, hence the kinematical one, by the constraint of a common origin; additional constraints may be added. Here again, the elementary steps consist in linear operations on vector and matrices of fixed dimension, allowing the user to easily proceed by successive trials and to optimize the strategy.

1 Introduction

1.1 What is a "trajectometer"?

Most of the particle detectors include layers which aim at recognizing and reconstructing as precisely as possible the trajectories of the charged particles produced in the main interaction or in a secondary vertex, usually curved by a magnetic field to provide an information on the momentum. These layers are thin (in terms of radiation lengths) to avoid disturbing the movement. They are generally located close to the interaction point (inner part of the detector for an implementation around a collider, or in first position within a fixed target experiment), before the calorimetric components. However some elements may be set at remote positions to provide a "lever arm" effect for a precise measurement of the curvature: this is the case of external muon detectors, which give both a signature of muons and a relatively precise measurement of their position. A schematic layout of a trajectometer is shown in Fig.1, in the case on planar layers; for a detector installed around the intersection region of a collider, a better description is obtained by replacing the planes by cylinders.

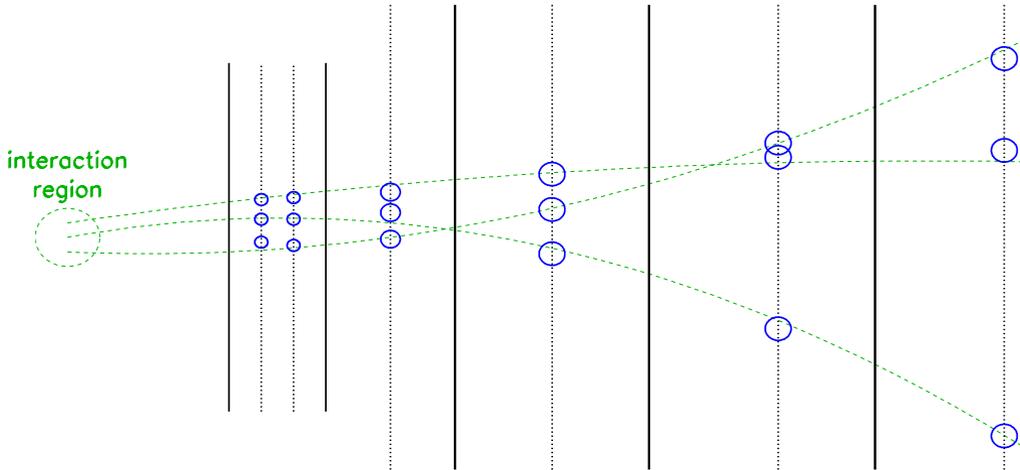


Figure 1. Schematic layout of a trajectometer. The green dashed lines are the trajectories of charged particles coming from the interaction region. The solid black lines are layers of material, the dotted black lines are measurement layers. The measurements are the blue circles (with a radius proportional to the uncertainty).

1.2 What is known and what is searched for?

The following properties of the detector are supposed to be known (or determined at a previous stage from calibration data):

- the equation of propagation, which determines the shape of the trajectories. In practice, it is defined by the magnetic field map within the trajectometer.
- the *nature* and the *precision* of the measurements (drift time, amplitude on signals in strips or pads, etc). In simple cases they just give one coordinate or a fixed combination of coordinates. They may also depend on the direction of incidence on the detection layer. In practice, what is needed is to express the measured quantity as a function of the local parameters of the trajectory (position and direction). The distribution of errors is generally gaussian, but possible exceptions should be evaluated and accounted for.
- the nature and the amplitude of the "noise" processes (random perturbations of the trajectory): the multiple Coulomb scattering, which modifies the direction, and the fluctuations of the energy loss, which affect the curvature. These perturbations depend on the amount and the nature of the material crossed by the particles, but also on their momentum. As a consequence, a good approximation of the momenta (or their inverse) has to be given by an external information (e.g. a calorimetric measurement), or by a first reconstruction where these perturbations are neglected.

In a first procedure (pattern recognition), a set of measurements is collected along the path of each particle. At this level there may be ambiguities or bad assignments, especially if there are many tracks, and even more if several interactions are "piled up" (this is the case in modern colliders). The next step (track fitting) may resolve some of these problems; actually, we will see that these two stages may be coupled into a mixed procedure including multiple trials and iterations. For the moment, we suppose that we have a clean collection of measurements along the trajectories of the charged particles. From this input, we want to extract the best estimation of:

- the properties of each particle (momentum, direction) at its origin. In practice, we want to obtain a backward extrapolation to the interaction region (the interior of the beam pipe for a collider), or to a secondary vertex of production. A magnetically curved trajectory may be described, when crossing a given reference surface (for example, for a given value of coordinate x), by 5 parameters: two for the position of the intersection with the reference surface (e.g. y, z for the example above) and 3 for the momentum vector (or the momentum and two angles). We suppose here that the mass is known or irrelevant; if not, several reconstructions may be needed, for different mass hypotheses.
- in some cases: a forward extrapolation towards external parts of the detector: e.g. the entry point into a calorimeter or into external muon chambers.
- the position and the composition of the primary vertex and secondary vertices within the detector, if any. This includes making one or several association of particles, and to use the constraint of a common origin to improve the reconstruction at the origin. Here again a procedure using iterative trials may be needed.

1.3 The problem of the noise

If the noise processes have negligible effects, we can choose a set of "initial" parameters \mathbf{p}_0 (position, direction, momentum) which gives a deterministic prediction of the expected measurements in layer k : $\bar{m}_k = F_k(\mathbf{p}_0)$. Because the measurement errors are independent we can write a χ^2 to be minimized:

$$\chi^2 = \sum_k \left(\frac{m_k - F_k(\mathbf{p}_0)}{\sigma_k} \right)^2$$

where m_k is the actual measurement, with an error σ_k . If needed (significantly non-gaussian errors), this may be replaced by a log-likelihood which is also a sum of independent terms. In most detectors the errors are small, so starting from a first approximation of the trajectory, F_k may be replaced by a linear function of the parameters and minimizing the χ^2 consists in solving a linear system of n_p equations, where n_p is the number of parameters. An iteration may be needed, but in general, the computation is fast.

The situation is more delicate if the errors induced by the noise are comparable to the measurement errors, or larger : a perturbation on the particle affects all measurements coming afterwards, so the measurements are no longer *independent*. The χ^2 should now be written as:

$$\chi^2 = \sum_{j,k} W_{jk} (m_j - F_j(\mathbf{p}_0))(m_k - F_k(\mathbf{p}_0))$$

where W is the inverse of the total covariance matrix, including the measurement errors and the noise induced errors, with non vanishing covariance terms: the measurements m_j and m_k are correlated through the perturbations occurring before both of them. This requires heavier computations; moreover, to make an optimal forward extrapolation we need to evaluate another covariance matrix, where m_j and m_k are correlated through the perturbations occurring *after* them.

In the following sections we apply to trajectometry an alternative method better suited to estimate the state of a dynamic system affected by random perturbations: the **Kalman Filter** (KF). The presentation differs slightly from the traditional one, but it is mathematically equivalent. Here we use systematically a *weight matrix* and *weighted mean* formalism, which has the advantage of being more "compact", and hopefully more user friendly, in the sense that it relies on various combinations of very few intuitively understandable elementary operations on matrices and vectors. We describe with

more or less details the operations within some frameworks (for example for some choices of parameters to define the trajectory), but we cannot give an exhaustive review of all possible implementations: the tools need to be adapted to a given context, after a clear understanding of the detector and the possible and useful approximations. In any case, the choices should be dictated by the gain in terms of the precision on physical quantities of interest for a further analysis.

2 The principle of the Kalman Filter (progressive fitting): a simple unidimensional example

2.1 The simplest model of measurement of a "noisy" process

We consider here a point with a random motion on a line. Its position $x(t)$ is measured without bias at times $1, 2, 3, \dots$, with independent measurement errors of variance σ^2 . The displacements between two measurements are 0 in average and independent, with the same variance τ^2 (for example, this is a brownian motion, as illustrated in Fig.2). The measurement errors are independent of the displacements.

Our problem is: if we have n successive measurements X_k of the true positions x_k , what is the combination of the X_k giving the best estimator of the initial position x_1 , or the final one x_n ? and why not, the best estimator of any intermediate position x_k ? The solution is simple if τ^2 is negligible (at any time: the average of the measurements), or if τ^2 is very large compared to σ^2 (the best for x_k is just X_k , because the other measurements are too much disturbed). It is less clear if τ^2 is comparable to σ^2 , or, more precisely, if $n\tau^2$ (the total variance of the displacements) is comparable to σ^2 : in that case, the cumulated displacements and the measurement errors cannot be disentangled.

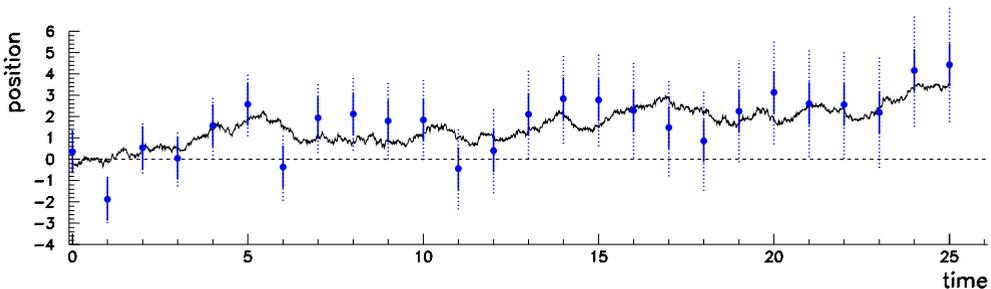


Figure 2. Measurements (blue points) of the position of a point moving randomly on a line. Here a brownian motion is simulated (solid line), and the variance of the displacement between two measurements is $\tau^2 = 0.25$. The solid bars represents measurement errors with $\sigma^2 = 1$; the dotted ones include the contribution of the displacements from the beginning, i.e. for point k it is $\sqrt{k\tau^2 + \sigma^2}$.

A first solution we may think about is the following: the difference between X_k and x_1 is the sum of $(k-1)$ independent displacements and one measurement error, so its variance is $V_k = (k-1)\tau^2 + \sigma^2$. We could make the *weighted average* of the X_k , with weights equal to $1/V_k$. This would be actually the best linear estimator, if the $X_k - x_1$ were *independent* random variables; but *this is not true* in our problem, because they include all displacements from the beginning: X_k and X_l both depend of the steps before k and l . As a consequence, if $l > k$, $cov(X_k, X_l) = k\tau^2$.

To build the best estimator in a standard way, we have to account for the $(n \times n)$ covariance matrix C of the X_k . A linear combination $\sum a_k X_k$ is an unbiased estimator of x_1 if $\sum a_k = 1$ and its variance is minimal if $\sum a_j a_k C_{jk}$ is minimal within the above constraint. That is, we have to solve a linear system of n equations; the solution may be written as a weighted mean of the X_k , with weights $w_k = \sum_j (C^{-1})_{jk} X_j$. The complexity of the problem grows more than linearly with n . Moreover, if we want to evaluate the final position, or an intermediate one, we have to build another covariance matrix and then solve a different linear system.

Fortunately, there is another way to obtain the same results, with a number of operations *proportional* to n through the **Kalman Filter** methodology [1, 4] (*progressive fitting* [2, 3]).

2.2 Tools of the progressive fitting

The fundamental idea is to incorporate the measurements *one after one* in the algorithm, in such a way that *independent* random variables are combined at any stage. The procedure is illustrated on Fig.3, which may help to understand the simple operations behind the mathematical formulae.

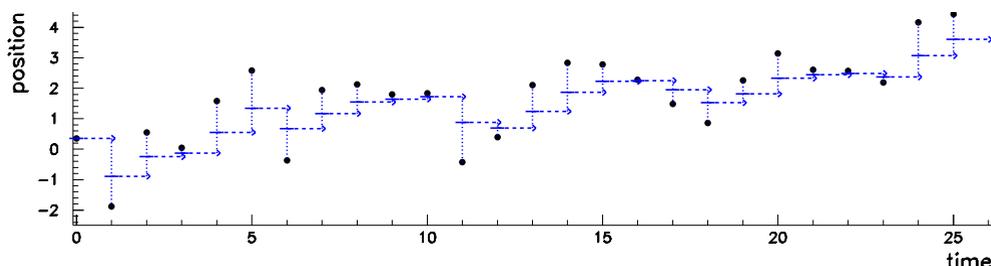


Figure 3. Principle of the forward filter, applied to the same dataset as in Fig2 (the black points are the measurements X_k of the successive positions x_k). The blue solid line at time k represents $\tilde{X}_{1 \rightarrow k,k}$, best estimator of x_k using the first k measurements (see text below), which is combined with the next measurement X_{k+1} to give $\tilde{X}_{1 \rightarrow k+1,k+1}$, best estimator of x_{k+1} , and so on.

Let us consider the following elementary steps:

- $X_1 = x_1 + \varepsilon_1 = x_2 - \eta_1 + \varepsilon_1$, where ε_1 is the measurement error on point 1 (variance σ^2) and η_1 the displacement from time 1 to time 2 (variance τ^2). Because ε_1 and η_1 are independent, we see that $X_1 - x_2$ has a variance $\sigma^2 + \tau^2$, that is, X_1 is equivalent to a measurement of x_2 with variance $\sigma^2 + \tau^2$.
- By definition, $X_2 = x_2 + \varepsilon_2$ is another measurement of x_2 , with variance σ^2 . The crucial point is that ε_2 is *independent* of both ε_1 and η_1 , so X_1 and X_2 are two *independent* measurements of x_2 .
- The "best" linear combination of two independent measurements (that is, with the lowest variance) is their weighted mean with weights equal to the inverse of their variance. Here, this means that the "best" combination of X_1 and X_2 to estimate x_2 is:

$$\tilde{X}_{1 \rightarrow 2,2} = \left(\frac{X_1}{\sigma^2 + \tau^2} + \frac{X_2}{\sigma^2} \right) / \left(\frac{1}{\sigma^2 + \tau^2} + \frac{1}{\sigma^2} \right) \quad \text{of variance} \quad \tilde{\sigma}_{1 \rightarrow 2,2}^2 = 1 / \left(\frac{1}{\sigma^2 + \tau^2} + \frac{1}{\sigma^2} \right)$$

(where the notation $\tilde{X}_{l \rightarrow m,k}$ means: best estimator of x_k using measurements l to m).

In the case of gaussian errors, the steps described above may be interpreted as operations on a likelihood function \mathcal{L} : including only the measurement X_1 of x_1 , \mathcal{L} is a gaussian function; accounting for the random displacement consists in a *convolution* with another gaussian function; combining two independent measurements is a *multiplication* of the corresponding gaussian functions. Of course, maximizing \mathcal{L} gives the same results as above. We will see later that this formulation can be extended to more complex cases.

The remarkable feature of this algorithm is that it can be iterated to include a further measurement (for convenience we replace $\tilde{\sigma}_{1 \rightarrow 2,2}$ by $\tilde{\sigma}$):

- $\tilde{X}_{1 \rightarrow 2,2}$ found above is a measurement of x_2 with variance $\tilde{\sigma}^2$, so it is a measurement of x_3 with variance $\tilde{\sigma}^2 + \tau^2$
- X_3 is another measurement of x_3 , with variance σ^2 , and $X_3 - x_3 = \varepsilon_3$ is independent of all random variables used to build $\tilde{X}_{1 \rightarrow 2,2}$.
- so we obtain the best linear combination of X_1, X_2, X_3 to estimate x_3 :

$$\tilde{X}_{1 \rightarrow 3,3} = \frac{\frac{\tilde{X}_{1 \rightarrow 2,2}}{\tilde{\sigma}^2 + \tau^2} + \frac{X_3}{\sigma^2}}{\frac{1}{\tilde{\sigma}^2 + \tau^2} + \frac{1}{\sigma^2}} \quad \text{of variance} \quad \frac{1}{\frac{1}{\tilde{\sigma}^2 + \tau^2} + \frac{1}{\sigma^2}}$$

Let us note that this may be written in a simpler way using the formalism of the *weight* (inverse of the variance) to express the combination as a *weighted mean*:¹

$$\tilde{X}_{1 \rightarrow 3,3} = \frac{\tilde{w}_{1 \rightarrow 2,2} \tilde{X}_{1 \rightarrow 2,2} + w_3 X_3}{\tilde{w}_{1 \rightarrow 2,2} + w_3}$$

The variance is $1/(\tilde{w}_{1 \rightarrow 2,2} + w_3)$, in other terms: the weight of the combined estimator is just the sum of the weights of the two independent estimators.

We can in the same way incorporate the fourth measurement, and so on, and build $\tilde{X}_{1 \rightarrow k,k}$ successively for all values of k . This is the "forward filter", which gives eventually the best estimator of the final position. It is clear that we can obtain the best estimator of the initial position through a similar formalism, starting from the last measurement and incorporating the other ones in reverse order. This is the "backward filter", giving, with our notations, $\tilde{X}_{k \rightarrow n,k}$ for any k . This is convenient if we have in mind a particle detector, where we are mainly interested to the initial parameters.

We can also build a χ_{\min}^2 associated to each step of the procedure, in the usual way:

- combining X_1 and X_2 to estimate x_2 , we have

$$\chi^2(X) = (X - X_1)^2/(\sigma^2 + \tau^2) + (X - X_2)^2/\sigma^2$$

then $\chi_{\min}^2 = (X_1 - X_2)^2/(2\sigma^2 + \tau^2)$

- at any further time $k + 1$ we combine $\tilde{X}_{1 \rightarrow k,k+1}$ ($\tilde{\sigma}_{1 \rightarrow 2,2}^2, \chi_{\min,k}^2$) with X_{k+1} . This gives a new value $\chi_{\min,k+1}^2 = \chi_{\min,k}^2 + (X_{k+1} - \tilde{X}_{1 \rightarrow k,k})^2/(\tilde{\sigma}_{1 \rightarrow 2,2}^2 + \sigma^2)$

As usual, in the gaussian case, $\chi_{\min}^2 = -2 \ln(\mathcal{L}_{\max})$, where \mathcal{L} is the likelihood, omitting a constant normalization factor, and it follows the law of χ^2 with $N - 1$ degrees of freedom, N being the number of measurements included.

The principle of one step of the filter is illustrated in Fig. 4.

¹ In the usual presentation of the KF, this step would be expressed as: find the best combination $\tilde{X}_{1 \rightarrow 2,2} + \lambda(X_3 - \tilde{X}_{1 \rightarrow 2,2})$, where λ is a "gain" coefficient. We prefer to use the weighted mean of independent variables, which will appear to be a universal tool for the next operations.

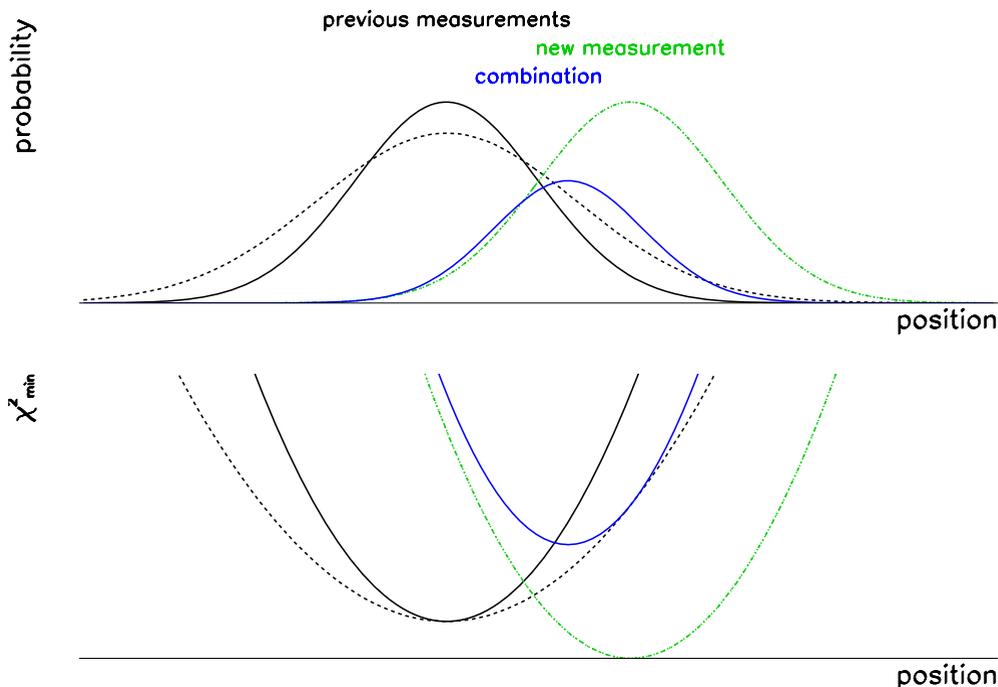


Figure 4. The estimator including all measurements up to a given point is represented in black (top: density of probability; bottom: minimum of χ^2), the dotted line accounts for the next random displacement. The green line represents the next measurement, and the blue one is the combination of these two independent estimators, which is the input for the next step.

2.3 Useful byproducts

We can build further tools with very few more efforts: it is easy to obtain the best estimator of any intermediate position using all measurements, that is, to build the "smoother" in the KF terminology (optimal *interpolation*), by an adequate combination of the forward and backward filters. The forward filter gives $\tilde{X}_{1 \rightarrow k, k}$ as an estimator of x_k with a variance that we call σ_f^2 , and the backward one gives $\tilde{X}_{k+1 \rightarrow n, k}$ with a variance $\sigma_b'^2$ (including a term τ^2 to go from $\tilde{X}_{k+1 \rightarrow n, k+1}$ to $\tilde{X}_{k+1 \rightarrow n, k}$); these estimators include independent measurement errors; the key point is that they also include *independent displacement errors* (the η_j with $1 \leq j \leq k$ for the forward one, $k < j \leq n$ for the backward one). So they can be simply combined as a weighted mean, defining $w_f = 1/\sigma_f^2$ and $w_b' = 1/\sigma_b'^2$:

$$\tilde{X}_{1 \rightarrow n, k} = \frac{w_f \tilde{X}_{1 \rightarrow k, k} + w_b' \tilde{X}_{k+1 \rightarrow n, k}}{w_f + w_b'}$$

The same result is obtained by combining $\tilde{X}_{1 \rightarrow k-1, p}$ and $\tilde{X}_{k \rightarrow n, k}$ with their proper variances (we just need to include X_k once and only once). We can also write an *exclusive* interpolation (without the measurement X_k) by combining $\tilde{X}_{1 \rightarrow k-1, k}$ and $\tilde{X}_{k+1 \rightarrow n, k}$: this will be useful for a discrimination of

possible outliers, as discussed below.

Up to now we have assumed a perfect situation: both the displacements and the measurement errors have the expected distribution. In practice, in particle detectors, there may be deviations due to rare phenomena affecting the trajectory (e.g. a hard scattering), bad measurements or bad assignments of points to a given trajectory (especially if many particles cross the same detector element). It is useful to build tools to detect abnormal deviations ("outliers") and to define a strategy to get rid of them, as far as possible. Within our very simple model we can define two kinds of tests which can be extended to realistic situations:

- for a given time k , the difference $\tilde{X}_{1 \rightarrow k,k} - \tilde{X}_{k+1 \rightarrow n,k}$ between the forward and the backward filters has a predicted standard deviation $\sqrt{\sigma_f^2 + \sigma_b^2 + \tau^2}$. If $\sigma_f^2 + \sigma_b^2$ is smaller or comparable to τ^2 , displacements which are large compared to τ may be detected; if not, the test can only discriminate very large deviations. If the distribution of the measurement and displacement errors are gaussian, a probability of χ^2 may be used as a discriminator.
- for time k , the measurement X_k and the exclusive interpolation (of variance σ_e^2) are independent, therefore the variance of their difference is $\sigma_e^2 + \sigma^2$. Here again, if σ_e^2 is not too large compared to σ^2 , outliers (abnormal measurements) may be detected. In the gaussian case, the probability of χ^2 gives a good discriminator.

Finally, let us remark that a large sample of clean measurements may be used to perform a *calibration* of the errors: if σ and τ are not known *a priori*, the distribution of the residuals mentioned above provide estimators for them, and possibly some hints on the distribution of errors.

2.4 Comments

The Kalman Filter was originally suited to dynamical problems like following the trajectory of engines from successive observations. In that case, the forward filter is the most natural tool: it can give in real time an updated estimation of the present state (position, direction, velocity). In applications to particle trajectometry, computations are not done in real time: even if some algorithms are implemented online, their input is a set of measurements coming after the particle has escaped the detector. Moreover the main purpose of the reconstruction of trajectories is to provide the best estimations at starting point (ideally, the vertex where this particle is originating from), so the backward filter is the basic tool. However, extrapolations to external detectors and interpolations may be needed, and discrimination of outliers is quite useful.

The number of operations to build the complete set of filters (forward, backward and smoother) is proportional to n if no outliers are removed. However, if a point k is to be removed after being compared to the interpolation, the forward filter has to be reevaluated at all points after k , and the backward filter at all points before k ; the smoother has to be redone everywhere.

3 More complex examples

In these examples, we want to go progressively to the description of a real detector. In particular, we do not consider measurements labeled by times, but measurements of one or several coordinates as functions of one coordinate describing the successive layers of the detector. To simplify some expressions we will use the following notations for matrices A , B and vectors \mathbf{q} :

$$A[\mathbf{q}] = \mathbf{q}^T A \mathbf{q} \quad A[B] = B^T A B$$

3.1 Straight line planar trajectory (2 parametres, linear model)

In this example (illustrated in Fig.5) the trajectory may be described as $y = ax + b$, and the coordinate y is measured as Y_k at x_k ($k = 1, \dots, n$), with a variance σ_k^2 . The noise consists in a random scattering (variation of the slope a) with a variance ρ_k^2 at each measurement layer x_k . All measurement and scattering errors are independent .

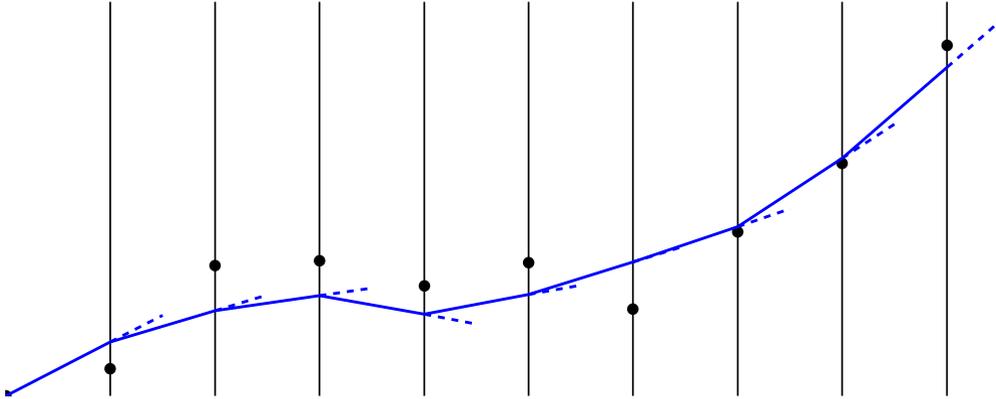


Figure 5. A planar trajectory made of straight line segments between measurement planes, where the slope is randomly modified (simple model for the multiple scattering on particles). The black points represent measurements.

The parameters to be fitted are a and b ; we call \mathbf{p} the vector (b, a) , or, more generally the vector of *local* parameters ($y, dy/dx$) at any point of the trajectory. Let C be the (2×2) covariance matrix of an estimator, and W its inverse (the *weight matrix*). If all errors are gaussian, the log-likelihood is a quadratic function of \mathbf{p} :

$$-2 \ln(\mathcal{L}) = -2 \ln(\mathcal{L})_{max} + W [\mathbf{p} - \tilde{\mathbf{p}}]$$

where $\tilde{\mathbf{p}}$ is the best estimator. In the general case, we will build the "best" linear estimator (using all measurements Y_k up to a given position) through linear combinations using the matrices C and/or W at different stages of a progressive fitting procedure.

Before that, we will try to analyze this problem of estimating (a, b) with the standard method, by computing the variance/covariance of the measurements. We call ε_k the error on Y_k and ζ_k the variation of slope at x_k . Then:

$$Y_k = ax_k + b + \varepsilon_k + \sum_{j < k} (x_k - x_j) \zeta_j$$

$$var(Y_k) = \sigma_k^2 + \sum_{j < k} (x_k - x_j)^2 \rho_j^2$$

² We suppose ρ_k to be known *a priori*. For a physics interpretation in terms of multiple scattering, this means that the momentum is measured elsewhere; this model is just an intermediate step towards a complete trajectometer, aimed to introduce the fundamental tools.

$$cov(Y_k, Y_l) = \sum_{j < \inf(k,l)} (x_k - x_j)(x_l - x_j)\rho_j^2$$

The best linear estimator is obtained by minimizing the global χ^2 :

$$\chi^2 = \sum_{j,k} W_{jk}(Y_j - ax_j - b)(Y_k - ax_k - b)$$

that is, we have to invert the $(n \times n)$ non-diagonal covariance matrix of the measurements.

Let us now describe one step of a progressive procedure; for convenience we will use *local* parameters (the value of y at x_k and the slope a). For the moment we suppose that we have built the best estimator $\tilde{\mathbf{p}}_{1 \rightarrow k,k}$ (matrices $\tilde{C}_{1 \rightarrow k,k}$, $\tilde{W}_{1 \rightarrow k,k}$) using $Y_1, Y_2 \dots Y_k$, associated the a minimum value $\tilde{\chi}_{1 \rightarrow k,k}^2$, and we want to build $\tilde{\mathbf{p}}_{1 \rightarrow k+1,k+1}$. We have to perform 3 operations:

- account for the scattering at x_k , by evaluating C for $\tilde{\mathbf{p}}_{1 \rightarrow k,k}$ as an estimator of the parameters *after* the scattering. In our model, we just need to account for an additional error on a , so we compute:

$$\tilde{C}'_{1 \rightarrow k,k} = \tilde{C}_{1 \rightarrow k,k} + \begin{pmatrix} 0 & 0 \\ 0 & \tau^2 \end{pmatrix} \quad \text{and} \quad \tilde{W}'_{1 \rightarrow k,k} = (\tilde{C}'_{1 \rightarrow k,k})^{-1}$$

The value of $\tilde{\chi}^2$ is not modified.

- *propagate* the estimator, going to the local parameters at x_{k+1} : we have $y_{k+1} = y_k + a(x_{k+1} - x_k)$, and the slope a is not modified. We write this simple transformation in a matrix formalism:

$$\begin{aligned} \tilde{\mathbf{p}}_{1 \rightarrow k,k+1} &= D_{k \rightarrow k+1} \tilde{\mathbf{p}}_{1 \rightarrow k,k} \quad \text{with} \quad D_{k \rightarrow k+1} = \begin{pmatrix} 1 & x_{k+1} - x_k \\ 0 & 1 \end{pmatrix} \\ \tilde{C}'_{1 \rightarrow k,k+1} &= \tilde{C}'_{1 \rightarrow k,k} [D_{k \rightarrow k+1}^T] \\ W'_{1 \rightarrow k,k+1} &= \tilde{W}'_{1 \rightarrow k,k} [D_{k+1 \rightarrow k}] \quad \text{using} \quad (D_{k \rightarrow k+1})^{-1} = D_{k+1 \rightarrow k} \end{aligned}$$

- *combine* $\tilde{\mathbf{p}}_{1 \rightarrow k,k+1}$ with the *independent* information given by Y_{k+1} . The combined χ^2 is:

$$\chi^2(\mathbf{p}_{k+1}) = \tilde{\chi}_{1 \rightarrow k,k}^2 + \tilde{W}'_{1 \rightarrow k,k+1} [\mathbf{p}_{k+1} - \tilde{\mathbf{p}}_{1 \rightarrow k,k+1}] + \frac{(Y_{k+1} - \tilde{y}_{1 \rightarrow k,k+1})^2}{\sigma_k^2}$$

We introduce now the 2-vector of *local measurement* $\mathbf{P}_{k+1} = (A, Y_{k+1})$ and its *weight matrix*

$$W_{k+1}^m = \begin{pmatrix} 1/\sigma_k^2 & 0 \\ 0 & 0 \end{pmatrix}$$

Actually A is not measured, but the expression of χ^2 does not depend on it; we can set it to 0 by convention. With these definitions we have to minimize:

$$\chi^2(\mathbf{p}_{k+1}) = \tilde{\chi}_{1 \rightarrow k,k}^2 + \tilde{W}'_{1 \rightarrow k,k+1} [\mathbf{p}_{k+1} - \tilde{\mathbf{p}}_{1 \rightarrow k,k+1}] + W_{k+1}^m [\mathbf{p}_{k+1} - \tilde{\mathbf{P}}_{k+1}]$$

The solution may be written as:

$$\tilde{\mathbf{p}}_{1 \rightarrow k+1,k+1} = (\tilde{W}'_{1 \rightarrow k,k+1} + W_{k+1}^m)^{-1} (\tilde{W}'_{1 \rightarrow k,k+1} \tilde{\mathbf{p}}_{1 \rightarrow k,k+1} + W_{k+1}^m \tilde{\mathbf{P}}_{k+1})$$

which is an extension of the weighted mean found in the simplest model (here the weights are matrices). We still have the property of *additivity of weights*: the weight matrix of the combination is the sum of the weight matrices of the independent estimators.

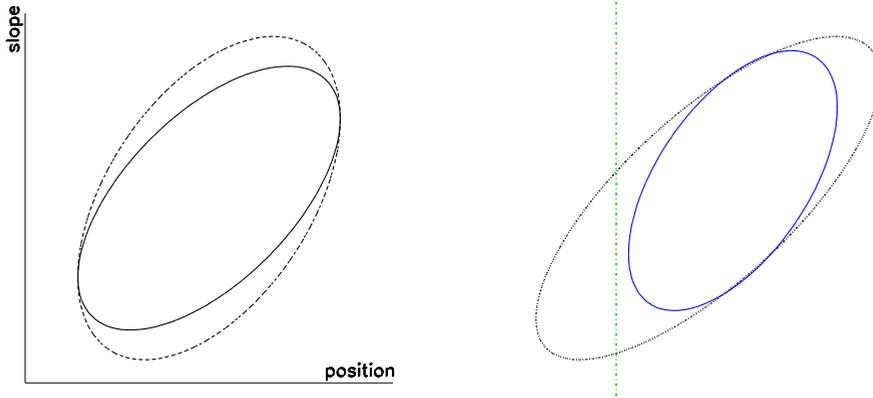


Figure 6. The operations of one step $k \rightarrow k + 1$ of the filter (as in Fig4) applied to a 2-parameter model (position and slope of a trajectory). The ellipses are the contours associated to one standard deviation around the central value. Left: solid: the "initial" estimator $\tilde{\mathbf{p}}_{1 \rightarrow k, k}$; dashed: including the scattering. Right: black, dotted: propagated to point $k + 1$; green, dash-dotted: measurement \mathbf{P}_{k+1} at point $k + 1$ (position only \rightarrow vertical strip); blue: combination.

The formalism above may be completely expressed with elementary operations on "atoms" of information (one "atom" = vector of local parameters + weight matrix + minimal χ^2), but we have evaded some technical problems for a practical coding of the algorithm, especially: how to begin this recursive procedure. With the first measurement we have not enough information to define both parameters: we have seen that we can manage this situation by taking a singular weight matrix $\text{diag}(1/\sigma^2, 0)$ and a measurement vector with an arbitrary value for the first element (0 for example): the χ^2 valley may be seen a strip in the (a, b) plane, along the a direction. This "atom" may be propagated to the next position with the formalism above: the valley is still infinite, by now in an oblique direction w.r.t. the local parameters; when combined to a strip in the a direction (local measurement) it results in a finite region, and both $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{W}}$ are regular.

The handling of the scattering remains to be clarified, because the operation $\tilde{\mathbf{C}}' = \tilde{\mathbf{C}} + \mathbf{C}^{scat}$ cannot be done if $\tilde{\mathbf{C}}$ is not yet defined (first point). However we can rewrite this operation as $\tilde{\mathbf{W}}' = (\tilde{\mathbf{W}}^{-1} + \mathbf{C}^{scat})^{-1} = (1 + \tilde{\mathbf{W}}\mathbf{C}^{scat})^{-1}\tilde{\mathbf{W}}$, which can be performed with one measurement only.

3.2 Further comments

The forward filter and the interpolators may be defined in the same way as in the simplest model. The tools to detect measurement outliers or abnormal scatterings may be built using χ^2 differences, with the same principles.

In the previous model the scattering occurred at the positions x_k where measurements were done. The formalism may be extended to any configuration: we just need to define the set of positions x_k where something happens (measurement or scattering) and to make the corresponding operations on $(\mathbf{p}, \mathbf{W}, \chi^2)$ by increasing x (forward filter) or decreasing x (backward filter), with propagation operations in between. In this way forward or backward extrapolations may be done to account for material

outside the range of measurements. An important application is the material between the vertex of origin and the first measurement (e.g. the beam pipe), to be accounted for in the analysis of the primary interaction.

A thick scattering region may be described, either as a set of elementary layers handled as above, or globally by computing the (2×2) matrix C^{scat} : at first order, the scattering through a thick material is fully described by terms to be added to the covariance matrix of y and a at a given reference position x_0 , including a non-diagonal one to account for their correlation.

3.3 Curved planar trajectory (3 parameters)

Here we want to introduce a good approximation of a detector making measurements in a plane xy perpendicular to a magnetic field, for trajectories with a moderate angle w.r.t. the x axis, and a large radius of curvature R compared to the size of the detector. In that case we can describe the trajectory with a linear function of 3 parameters a, b, c :

$$y = ax + b + x^2/2R = ax + b + cx^2/2$$

We assume as above that y is measured at x_k ($k = 1, 2, \dots, n$) with a variance σ_k^2 . The 3-vector of local parameters is $\mathbf{p} = (y, dy/dx, d^2y/dx^2)$. In this model we can account for scatterings (random variations of dy/dx) and also for both deterministic and random variations of energy, which are expressed as variations of the curvature d^2y/dx^2 . The formalism of the filters is exactly the same as in the previous model (plus a specific operation to modify the curvature parameter in case of energy loss), with a few technical differences:

- The matrix W is regular only when at least 3 points are included in the fit. In practice, it is of rank 1 with one point (the χ^2 valley is a slice in 3D space), 2 with 2 points (the χ^2 valley is a tube). But the same formalism as above may be used: a measurement is represented by a vector $(Y_k, 0, 0)$ with weight matrix $diag(1/\sigma^2, 0, 0)$ and in the initial vector of parameters undefined values are set to 0.
- In this model the momentum may be estimated from the fit itself so the variance of the scattering angles may be computed without external information; more precisely: the relevant curvature parameter c is proportional to the inverse of the momentum, with a geometrical sign depending on the physical sign of the charge, which is also to be determined. However, the curvature is not supposed to be known at the beginning of the filter. As a consequence, an iteration is needed: for example, the trajectory is fitted first ignoring the scattering, and the curvature found is injected in a second pass; if the curvature is significantly modified, a third pass may be needed. If the measurement range is too short to provide a significant estimation of the curvature, an external information is needed to use the noise formalism in the filters.

3.4 Realistic trajectory in space: using a linear approximation

In real detectors, no linear model (as the parabolic one) may represent the trajectories with the accuracy requested by the precision of the measurements. However, if the magnetic field is regular, one can choose initial parameters such that the position and the direction of the trajectory in any measurement layer depends smoothly on them³. In these conditions a *reference trajectory* determined by

³this may be wrong for low energy particles at the end of their range, but in that case the contribution of the noise is so large that the precision of the measurement is not really significant, and a precise prediction of the trajectory is not needed.

initial parameters \mathbf{p}_0 may be defined as a first approximation, such that the functions F_k introduced in Sect.1.3 may be replaced by a *linear expansion*:

$$F_k(\mathbf{p}_0 + \delta\mathbf{p}) = F_k(\mathbf{p}_0) + (\nabla F)_0 \cdot \delta\mathbf{p}$$

Let us take an example to illustrate a practical application of this method (and possible limitations): a circular trajectory in a (xy) plane (see Fig. 7).

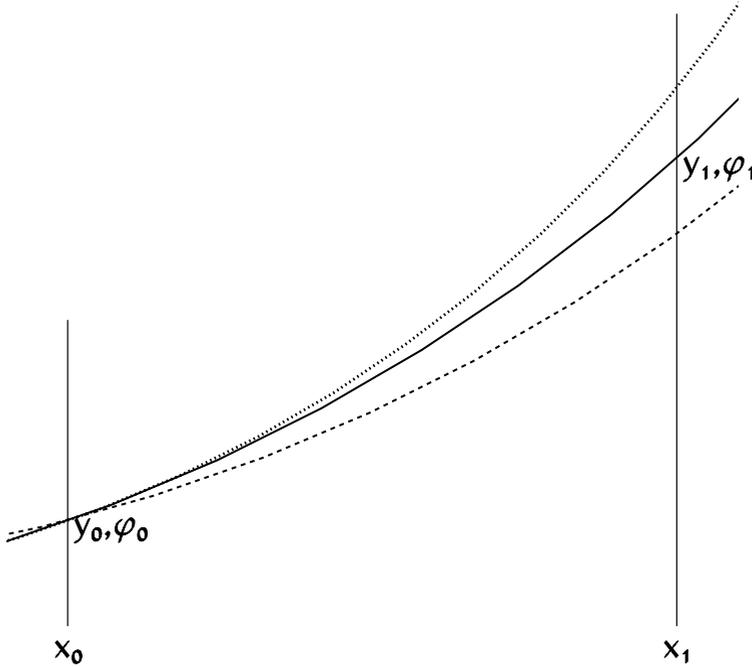


Figure 7. A planar circular trajectory measured at fixed x . Solid: the reference trajectory; dashed: with a variation of the initial direction φ_0 ; dotted: with a variation of the curvature c (changing the initial position \mathbf{p}_0 gives a simple translation). The parameters at x_1 (y_1, φ_1) depend linearly on small variations of the initial parameters at x_0 .

The initial parameters (at $x = 0$) are $y_0, \varphi_0, c = 1/\mathcal{R}$, where φ is the local direction ($\tan \varphi = dy/dx$) and \mathcal{R} the radius R with a *geometrical sign*: by convention we take + for an anticlockwise trajectory; in this model c is constant. With our convention we can write:

$$x = x_0 + \mathcal{R}(\sin \varphi - \sin \varphi_0) \qquad y = y_0 + \mathcal{R}(\cos \varphi - \cos \varphi_0)$$

First we want to evaluate the parameters y_1, φ_1, c at a fixed abscissa x_1 . The first equation gives two solutions for φ_1 (or no one !), and we have to choose one of them. In general it is the nearest one to φ_0 (but an actual measurement can correspond to the second solution if it is within the detector...). Once the “right” solution (y_1, φ_1) is found, we want compute the derivatives of y_1, φ_1 w.r.t. y_0, φ_0, c . Differentiating the first equation we obtain the derivatives of φ_1 (with $\Delta x = x_1 - x_0$):

$$d\varphi_1 = \frac{\cos \varphi_0}{\cos \varphi_1} d\varphi_0 - \frac{\Delta x}{\mathcal{R}^2 \cos \varphi_1} d\mathcal{R} \quad \rightarrow \quad \frac{\partial \varphi_1}{\partial \varphi_0} = \frac{\cos \varphi_0}{\cos \varphi_1} \quad ; \quad \frac{\partial \varphi_1}{\partial c} = \frac{\Delta x}{\cos \varphi_1}$$

Let us now differentiate the second one:

$$dy_1 = dy_0 - (\cos \varphi_1 - \cos \varphi_0)dR + \mathcal{R}(\sin \varphi_1 d\varphi_1 - \sin \varphi_0 d\varphi_0)$$

Injecting the expression of $d\varphi_1$ found above in this equation, we can extract after some algebra the derivatives of y_1 :

$$\frac{\partial y_1}{\partial y_0} = 1 \quad \frac{\partial y_1}{\partial \varphi_0} = \frac{\sin(\varphi_1 - \varphi_0)}{c \cos \varphi_1} \quad \frac{\partial y_1}{\partial c} = \frac{1 - \cos(\varphi_1 - \varphi_0)}{c^2 \cos \varphi_1}$$

It is interesting to note that when R is large, c is a more convenient parameter than R or \mathcal{R} : in that case $\varphi_1 - \varphi_0$ is small and proportional to c , so all derivatives go to a finite limit when $R \rightarrow \infty$; moreover, the value $c = 0$ (straight line, infinite R) is not a singularity, and the geometrical sign may freely change during a progressive fit.

These expressions give us the expression of the 3×3 propagation matrix D similar to the 2×2 matrix defined in Sect.3.1. We give in parentheses the approximation for weakly curved trajectories (small $\Delta\varphi$), introducing the length of the arc between the two points $\ell = \Delta\varphi/c$

$$D = \begin{pmatrix} 1 & \frac{\sin(\Delta\varphi)}{c \cos \varphi_1} (\simeq \frac{\ell}{\cos \varphi_1}) & \frac{1 - \cos(\Delta\varphi)}{c^2 \cos \varphi_1} (\simeq \frac{\ell^2}{2 \cos \varphi_1}) \\ 0 & \frac{\cos \varphi_0}{\cos \varphi_1} (\simeq 1) & \frac{\Delta x}{\cos \varphi_1} (\simeq \ell) \\ 0 & 0 & 1 \end{pmatrix}$$

This formalism should be used with care when the trajectory is close to a real singularity (quasi tangent to the measurement layer, that is $\cos \varphi_1 \simeq 0$): then the linear approximation is no longer valid, and moreover the actual meaning of the measurement may not be the coordinate y , and depend on the internal structure of the detection layer. In such a situation, it may help to redefine the parameters (e.g. take x at fixed y) and to express specifically the response of the detector under a skimming incidence; if this is not possible, the best solution is to ignore the measurement.

Once a convenient parametrization and a reference trajectory are found, the linear formalism using weight matrices may be applied to the vector $\delta\mathbf{p}$. The C and W matrices are the same for \mathbf{p} and $\delta\mathbf{p}$. If the deviations from the reference are too large, it may be iteratively redefined until a satisfactory one is found. It is also possible to use different references for different parts of the trajectory; to go from one part to the next one, the parameters and their weight matrix need to be transformed through an operation similar to the propagation described in Sect. 3.1 (see below).

3.5 Convenient parameters in usual detector configurations

We consider here two main categories of detectors: fixed target experiment or collider. In the first case the detection layers are mainly planes perpendicular to the beam axis (z coordinate) in the forward region, and possibly planes parallel to the beam around the target; in the second one there is a ‘‘barrel’’ part (cylinders of axis along z) and endcaps (planes perpendicular to z). Other configurations are possible, for example with ‘‘oblique’’ layers; this will be discussed later. If there is a magnetic field, we will use S/p to describe the curvature (p is the momentum, S the physical sign). In some cases (e.g. roughly uniform field along z) it is more convenient to use S/p_t (p_t is the transverse momentum).

3.5.1 Cartesian parameters

When the detectors are planes (e.g. at fixed z), a natural choice is x, y to describe the position within the plane, two slopes ($u = dx/dz, v = dy/dz$) or two angles to describe the local direction; for example a polar angle θ w.r.t. the z axis, and a azimuthal angle φ in projection onto the xy plane.

3.5.2 Cylindrical parameters

If the detection layers are cylinders around the beam axis, cylindrical coordinates (r, Φ, z) are natural parameters for the position: more precisely, the position at fixed r (in a detector layer) is defined by Φ, z (optionally $r\Phi, z$ for homogeneity). The direction may be given by θ, φ ⁴. If the field is uniform and parallel to z axis, the trajectory is a helix of radius R . As in Sect. 3.4, we use \mathcal{R} with a geometrical sign (+ if the trajectory is anticlockwise in xy projection). Using a point x_0, y_0, z_0 on the trajectory and φ as a running parameter, the trajectory is defined by:

$$\begin{aligned} x &= x_0 + \mathcal{R}(\sin \varphi - \sin \varphi_0) = r_0 \cos \Phi_0 + \mathcal{R}(\sin \varphi - \sin \varphi_0) \\ y &= y_0 - \mathcal{R}(\cos \varphi - \cos \varphi_0) = r_0 \sin \Phi_0 + \mathcal{R}(\cos \varphi - \cos \varphi_0) \\ z &= z_0 + \mathcal{R} \cot \theta (\varphi - \varphi_0) \end{aligned}$$

3.5.3 The “perigee” parameters

It may be useful to summarize the information about the trajectory in one set of intrinsic parameters instead of using an arbitrary reference surface. In the case of quasi-uniform magnetic field along the beam axis (by convention the z axis), we can use the “perigee”, point of closest approach to the z axis: if the particle originates from the main vertex, this point will be close to this vertex, so it will give a good approximation of the particle momentum. Another advantage is that a propagation of the trajectory and its error matrix to this point includes most of the material actually crossed by the particle (all material if the perigee is within a vacuum region, e.g. the beam pipe), so if this material is taken into account properly, the perigee parameters may be used in a further step of vertex fitting in a purely geometrical way, without accounting for noise: this will be exploited in Sect. 6.

The trajectory is defined by 5 parameters (see Fig. 8): the cylindrical coordinates of the perigee $(\varepsilon, \Phi_p, z_p)$, the signed curvature $c = 1/\mathcal{R}$ and θ . To avoid discontinuities around the origin when extrapolating a trajectory towards the interaction region, it is convenient to give a geometrical sign to ε : by convention it is positive if the origin O is on the right hand side of the trajectory, and Φ_p is defined as $\varphi_p + \pi/2$. With this convention, we have always $x_p = \varepsilon \cos \Phi_p$, $y_p = \varepsilon \sin \Phi_p$, and the trajectory may be parametrized as:

$$\begin{aligned} x &= \varepsilon \cos \Phi_p + \mathcal{R}(\sin \varphi - \sin \varphi_p) = \varepsilon \cos \Phi_p + \mathcal{R}(\sin \varphi + \cos \Phi_p) \\ y &= \varepsilon \sin \Phi_p - \mathcal{R}(\cos \varphi - \cos \varphi_p) = \varepsilon \sin \Phi_p - \mathcal{R}(\cos \varphi - \sin \Phi_p) \\ z &= z_p + \mathcal{R} \cot \theta (\varphi - \varphi_p) = z_p + \mathcal{R} \cot \theta (\varphi - \Phi_p + \pi/2) \end{aligned}$$

In the vertex fitting procedure, we need in principle short range extrapolations from the perigee, so we can use the second order approximation in $\ell = \mathcal{R}(\sin \varphi - \sin \varphi_p)$ (distance from perigee in xy projection):

$$\begin{aligned} x &= \varepsilon \cos \Phi_p + \ell \sin \Phi_p + c \ell^2 \cos \Phi_p \\ y &= \varepsilon \sin \Phi_p - \ell \cos \Phi_p + c \ell^2 \sin \Phi_p \\ z &= z_p + \ell \cot \theta \end{aligned}$$

The perigee parameters will also be used as a technical tool to compute the derivative matrices needed to propagate the error matrix in cylindrical coordinates (see Appendix).

⁴ To avoid confusions it is important to use different notations for Φ and φ for these two independent parameters.

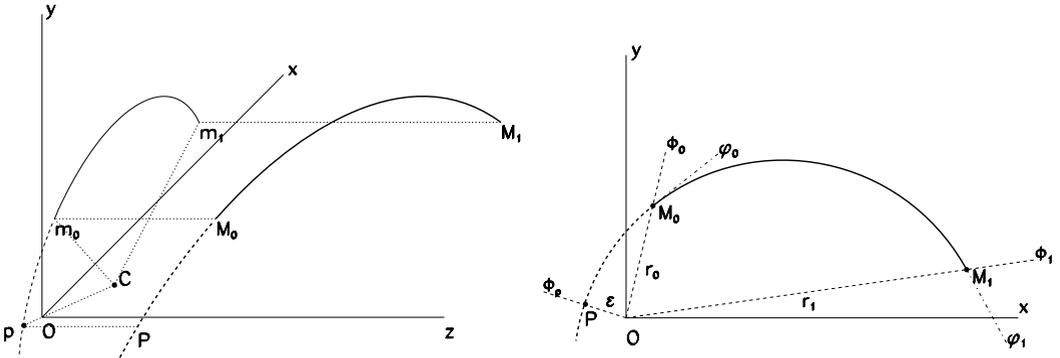


Figure 8. Left: the trajectory in 3D-space (helix of axis along z). Solid: the measured portion M_0M_1 , dashed: the extrapolation to the perigee P . Right: The projection onto the xy (circle). The position of a point along the trajectory is defined by r, Φ ; the direction of the tangent is defined by φ . In this example, the signed radius \mathcal{R} is negative (φ decreases with increasing r), and the perigee distance ε is positive.

3.5.4 Propagation of error matrices

In the helix model for trajectories, the analytical computation of the matrix of derivatives D was done in Sect.3.4 for cartesian parameters. Using similar techniques, one can obtain analytical expression for cylindrical parameters, as function of the parameters at the initial and the final point. The computation is developed in Appendix.

If the magnetic field is not perfectly uniform, the trajectory has to be propagated with a precision better than the measurements, so a numerical computation (or a perturbative expansion) may be needed. However, the derivatives may be taken from the analytical expressions, because they give a sufficient approximation for the propagation of errors.

3.5.5 Local change of parametrization

To follow the disposition of the detector layers, it may be convenient to modify the parametrization at a certain point of the trajectory. For example, with the helix model in cartesian coordinates, we use parameters $(x, y, \theta, \varphi, c)$ at fixed z in the forward region, and $(x, z, \theta, \varphi, c)$ at fixed y in the lateral region. Here using the same notation x for a parameter with a different meaning may be a source of confusion: the transformation of the error matrix should account for a non trivial transformation on the position parameters: an elementary variation δy of y in a plane $z = z_0$, at given values of x, θ, φ, c , is a translation which results in a displacement of the intersection of the trajectory with a plane at fixed y ; using the notation $a|_b$ for “ a at fixed b ”, and noting \mathbf{u} the unit vector along the trajectory, the variations of the coordinates $x|_y, z|_y$ are:

$$\delta x|_y = -\frac{u_x}{u_y} \delta y|_z = -\cot \varphi \delta y|_z \quad ; \quad \delta z|_y = -\frac{u_z}{u_y} \delta y|_z = -\frac{\cot \theta}{\sin \varphi} \delta y|_z$$

For the same reason, if the curvature is not negligible, a variation of $y|_z$ will affect the direction (actually, only φ) at fixed y ; if we call $\delta \ell$ the displacement in xy projection, we have from the helix

model:

$$\delta\varphi = c \delta\ell = -\frac{c}{\sin\varphi} \delta y$$

The jacobian matrix of the transformation from $(x, y, \theta, \varphi, c)|_z$ to $(x, z, \theta, \varphi, c)|_y$ is then:

$$D_{\text{loc}} = \begin{pmatrix} 1 & -\cot\varphi & 0 & 0 & 0 \\ 0 & -\frac{\cot\theta}{\sin\varphi} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{c}{\sin\varphi} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The covariance and the weight matrix are modified as $C^{\text{new}} = C^{\text{old}} [D_{\text{loc}}^T] = D_{\text{loc}} C^{\text{old}} D_{\text{loc}}^T$ and $W^{\text{new}} = W^{\text{old}} [D_{\text{loc}}^{-1}] = (D_{\text{loc}}^{-1})^T W^{\text{old}} D_{\text{loc}}^{-1}$.

For any other change of local parameters, a similar study has to be done to define the jacobian matrix.

3.6 Indirect measurements of parameters ("oblique" projection)

Up to now we have represented a layer of by a simple surface (e.g. a plane of wires). The quantity actually measured by a detector is supposed to depend on the position of the intersection of the trajectory with this surface, but it may also depend on the direction of incidence. Let us take two examples:

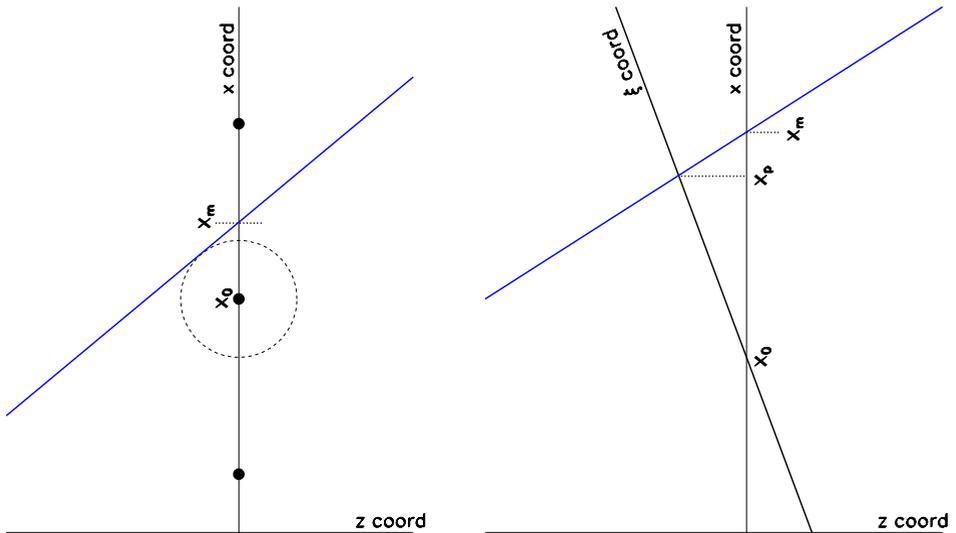


Figure 9. Examples of “oblique” measurements. The trajectory (blue line) has a slope $a = dx/dz$. In both cases, x_m represents the coordinate effectively measured in the plane $z = z_0$ through a raw measurement in the detector. Left: the raw measurement is the distance to a line parallel to y axis (e.g. a wire). Right: the raw measurement ξ is taken in a detector inclined by α (thick black line); we see on the figure that $x_p - x_0 = \xi \cos \alpha$ and $x_m - x_p = a \xi \sin \alpha$.

- In a plane $z = z_0$, we measure the distance of closest approach of the trajectory to a wire at $x = x_0$ (see Fig.9, left). Using the parameters x and $a = dx/dz$, and assuming that the curvature is negligible at this scale, this means that we measure $d = |x - x_0|/\sqrt{1 + a^2}$, with a precision σ . Within a good approximation, we can take the reference value a_{ref} of the slope, and consider that we measure $x = x_0 \pm d\sqrt{1 + a_{\text{ref}}^2}$ with a precision $\sigma_x = \sigma\sqrt{1 + a_{\text{ref}}^2}$ (at this level there may be an ambiguity if the extrapolation provided by the filter is not precise enough).
- The detector surface is not exactly perpendicular to z axis (see Fig.9, right). For example, we measure a coordinate ξ in a plane inclined by α on the xy plane, intersecting the plane $z = z_0$ at $x = x_0$ ($\xi = 0$ at the intersection). We obtain $x - x_0 = (\cos \alpha + a_{\text{ref}} \sin \alpha)\xi$. Here again we apply to the measurement a factor depending on the local direction.

The real situation may be more complex. For example, in a drift chamber, the relation between the measured time and the local parameters depends on the position (close to the wire or far away). Or we may have to consider in a barrel detector (with cylindrical parameters) detector elements which are planar. In any case, the prescription is to write the local parameter to be measured as a linear function of the quantity which is actually measured, with coefficients depending on the local direction of the trajectory.

3.7 Composite measurements

Some detectors (e.g. chambers with tilted wires) provide a measurement of a “composite” coordinate, e.g. a quantity $u = \alpha x + \beta y$ at fixed z , measured with a precision σ . The formalism of “atoms” introduced in Sect.3.1 is very convenient to account for such measurement u_m : it is equivalent to a vector $\mathbf{P} = (x_m, y_m, \dots)$, where x_m, y_m are any values such that $u_m = \alpha x_m + \beta y_m$, the other components being arbitrary, with a weight matrix of rank 1 (written here with 5 parameters):

$$W = \frac{1}{\sigma^2} \begin{pmatrix} \alpha^2 & \alpha\beta & 0 & 0 & 0 \\ \alpha\beta & \beta^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

More generally, let us suppose that we measure in a detector surface a set of n quantities $\mathbf{U} = (u_1, u_2 \dots u_n)$ ($n \leq 5$) that can be expressed locally as a linear combination of the $\delta\mathbf{p}$:

$$\mathbf{U} = \mathbf{U}_0 + M\delta\mathbf{p}$$

The errors on the measurements of $u_1, u_2 \dots u_n$ may be independent or not. Let us call C_U their covariance matrix, and $W_U = C_U^{-1}$ their weight matrix. We can introduce in the filter formalism an “atom” made with $\delta\mathbf{p}_m$ (any values compatible with the n measurements) and a weight matrix $W_m = M^T W_U M$ of rank n . The result of the weighted means does not depend on the arbitrary choices made to build \mathbf{p}_m .

3.8 Exogenous measurements

We call “exogenous” a measurement coming from a detector which does not belong to the trajectometer, or an information coming from an element of the trajectometer, but of a different nature. In the first category we can include a calorimetric measurement of the energy (if a matching can be established):

this may be useful to compute an initial curvature parameter when starting the backward filter (but the ambiguity on the sign has to be solved). In the second category we may have a timing information, or an evaluation of the ionization rate, that can constrain the momentum, or solve the mass ambiguity. When written as a linearizable function of the local parameters, these measurements can be handled in the same way as the composite measurements above.

3.9 Comments on practical implementation

A big advantage of the Kalman Filter formalism is to rely on linear operations on vectors and matrices of *fixed dimension* (number of parameters needed to describe the trajectory), whatever the number of measurements and noise sources. The implementation is computationally efficient if these operations are explicitly coded, without calling functions from a general matrix package. Moreover useful approximations may result in sparse matrices, reducing even more the computations needed. As a consequence, such procedures could even be introduced in high level triggers.

4 Coupling the pattern recognition to the track fit

In the previous section, we have supposed that the measurements to be affected to a given trajectory were unambiguously defined in a previous step of *pattern recognition*. In practice, for a complex event including many particles, this preliminary task is far from being easy, and in most cases it cannot be achieved without any ambiguity. The progressive fitting procedure can help to resolve these ambiguities, for example using the probabilities of χ^2 . For a better discriminating power, it may also be used within the pattern recognition procedure itself, to perform a progressive collection of points along the trajectory. The basic procedure [5, 6] is as follows:

- build tentative “segments” using points from a few layers, with loose criteria of compatibility; in this step ambiguities are freely accepted.
- apply a forward and/or backward filter to these segments.
- extrapolate to the next and/or previous layer and try to add a measurement found in this layer, and apply a χ^2 criterion to accept or reject this measurement; at this level ambiguities are still accepted (and possibly extended if several measurements are compatible with the extrapolation).
- iterate the procedure. In principle the χ^2 is more and more selective with more and more points included.
- at the end, resolve the remaining ambiguities if any (or keep some of them open for a final analysis).

In any case, the strategy should be adapted to the context on the following points: choosing the best starting region, tuning the criteria at each step, defining tolerance for missing points, using approximations in the filter (for example: ignoring the noise, assuming a small curvature), etc. In some cases, an external measurement (e.g. from a calorimeter or a muon chamber) may provide a good starting segment; if the first layers are very precise (as in usual “vertex detectors”), it can be used to define clean segments to be extrapolated forwards, because at this level there are few parasitic tracks produced in the material, and the trajectories are quasi straight lines. Hybrid strategies may also be efficient; there is no general rule on this subject.

5 Beyond the gaussian approximation

The measurement errors and the perturbations on the trajectory are never exactly gaussian. Some deviations from “gaussianity” are not worrying, because the convolution of different errors tend to

“gaussianize” the combination. For example, let us imagine a series of hodoscopes which just provide an interval (x_1, x_2) for a coordinate x at different position in y along a straight trajectory in xy plane, described by parameters a, b : in the absence of multiple scattering, each measurement gives slice in the a, b plane, and the global information is a polygon which is more or less extended depending on the position of the trajectory, while the gaussian model gives an ellipse with a shape depending only on the coordinates y_k of the hodoscopes; on the contrary, accounting for the multiple scattering results in a smoothing of the distribution of errors. Modern detectors are generally not hodoscopes, and the distribution of errors is often smooth and nearly gaussian. In the case of precise measurements, the non-gaussianity is wiped out by the “noise” along the trajectory.

More serious is the problem of errors with long tails, especially in the energy loss of electrons or positrons, which may be large even through a moderate amount of matter. These tails are propagated throughout the fitting procedure, so that the fitted values do not follow a gaussian distribution, and their variance is underestimated in the gaussian model. We have described above tools to detect abnormal deviations, but we want to go further and try to use explicitly the shape of the error distribution in the case where it is known, or predictable from the parameters of the trajectory. In practice we have to find a reasonable compromise between an *ideal* procedure (complete description and propagation of the errors), which will appear to be extremely heavy with several parameters, and the available computing power; we also want to have an idea of what we can gain with respect to the gaussian procedure, which is very fast.

5.1 The ideal procedure

The fitting procedure is still a forward or backward chain of basic operations (measurement, noise, propagation) along the trajectory, but now the “atom” of information is a density function $F(\mathbf{p})$ in the space of parameters, which express the likelihood of the subset of measurements included from the beginning of the chain. The previous considerations on the *independence* of the errors are still valid, so the mathematical transformations of F corresponding to the basic operations are:

- **measurement**: combination of independent informations, that is a **product**:

$$F^{\text{new}}(\mathbf{p}) = F^{\text{old}}(\mathbf{p}) f^{\text{meas}}(m(\mathbf{p}))$$

where m is the expression of the local measurement as a function of the local parameters, and f the distribution of the error on m .

- **noise**: addition of independent errors, that is a **convolution**:

$$F^{\text{new}}(\mathbf{p}) = F^{\text{old}}(\mathbf{p}) * g^{\text{noise}}(\mathbf{p})$$

- **propagation**: going from one layer to the next one consists in a transformation of the local parameters, that is a **composition**:

$$F^{\text{new}}(\mathbf{p}) = F^{\text{old}}(\mathcal{P}^{-1}(\mathbf{p}))/J(\mathcal{P}) \rightarrow F^{\text{new}} = F^{\text{old}} \circ \mathcal{P}^{-1}/J(\mathcal{P})$$

where \mathcal{P} is the transformation from the local parameters in the initial layer to the local parameters in the final one, and $J(\mathcal{P})$ its jacobian determinant.

None of these operations can be performed in a reasonable computing time in a multidimensional space (5 parameters in the standard implementation), without an adequate parametrization of F , f^{meas} and g^{noise} .

5.2 The Gaussian Sum Filter

One practical solution is to replace all functions involved in the different steps by a *sum of gaussian functions* [8]. The main advantage is that such functions are defined by a small set of values (the mean \mathbf{p}_0 and the weight matrix W); both their product and their convolution are gaussian, and the mean value and the weight matrix of the result have simple expressions. We summarize here the algebra of gaussian functions in a N -dimensional space:

$$\begin{aligned}
 \text{normalized density } G_{\mathbf{p}_0, W}(\mathbf{p}) &= \sqrt{\frac{\det(W)}{(2\pi)^N}} \exp\left(-\frac{W[\mathbf{p} - \mathbf{p}_0]}{2}\right) \\
 \text{product } G_{\mathbf{p}_1, W_1} G_{\mathbf{p}_2, W_2} &= C_{12} G_{\bar{\mathbf{p}}, W_1 + W_2} \\
 \text{with } \bar{\mathbf{p}} &= (W_1 + W_2)^{-1}(W_1\mathbf{p}_1 + W_2\mathbf{p}_2) \quad (\text{weighted mean of } \mathbf{p}_1 \text{ and } \mathbf{p}_2) \\
 \text{and } C_{12} &= \sqrt{\frac{\det(W_1) \det(W_2)}{(2\pi)^N \det(W_1 + W_2)}} \exp\left(-\frac{W_1[\mathbf{p}_1 - \bar{\mathbf{p}}] + W_2[\mathbf{p}_2 - \bar{\mathbf{p}}]}{2}\right) \\
 \text{convolution } G_{\mathbf{p}_1, W_1} * G_{\mathbf{p}_2, W_2} &= G_{\mathbf{p}_1 + \mathbf{p}_2, (W_1^{-1} + W_2^{-1})^{-1}}
 \end{aligned}$$

In the linear approximation, the propagation may be expressed as in Sect. 3: when going from \mathbf{p}_i to \mathbf{p}_f , with a jacobian matrix $D_{i \rightarrow f} = \partial \mathbf{p}_f / \partial \mathbf{p}_i$, we obtain:

$$W_f = W_i [D_{i \rightarrow f}^{-1}] = W_i [D_{f \rightarrow i}]$$

If we can approximate the measurement and the noise density functions as linear combinations of normalized gaussians, with *positive* coefficients:

$$f^{\text{meas}} = \sum_j a_j G_j \quad g^{\text{noise}} = \sum_k b_k G_k \quad \text{with} \quad \sum_j a_j = \sum_k b_k = 1$$

we obtain at each step of the procedure F as a combination of gaussian terms, which is automatically *positive* in the whole space of parameters. Of course, the main problem is that after n steps including each a sum with m_i coefficients, F is expressed as a sum of $\prod m_i$ terms, so the complexity may be too high if the detector has many layers. This can be partly cured by reducing the number of terms after each step, for example, suppressing the terms with low coefficients, or grouping similar terms into one. The strategy should be tuned for a given detector configuration.

To illustrate the method and the possible gain, we come back to our simplest model with one parameter (Sect.2.1), with one difference: the displacement η between two measurements is no longer gaussian. We adopt here an asymmetric superposition of gaussian functions, with mean value 0:

$$g(\eta) = \frac{a_1 G_{\mu_1, \tau_1} + a_2 G_{\mu_2, \tau_2} + a_3 G_{\mu_3, \tau_3}}{a_1 + a_2 + a_3} \quad \text{with} \quad a_1\mu_1 + a_2\mu_2 + a_3\mu_3 = 0$$

The variance of the displacement is then $\tau^2(\eta) = (a_1(\mu_1^2 + \tau_1^2) + a_2(\mu_2^2 + \tau_2^2) + a_3(\mu_3^2 + \tau_3^2)) / (a_1 + a_2 + a_3)$. In the following, we take for the triplets (a, μ, τ) : (10, -1, 0.3), (3, 0, 3) and (1, 10, 10), which give $\tau(\eta) = 4.122$; the measurements are gaussian with variance 1. We perform a series of trials of 6 measurements with 5 intermediate displacements; we apply to each sample the standard gaussian filter and the gaussian sum filter (keeping all 3^5 terms), to find an estimator of the initial position. The gaussian sum may be used through its mean value and its standard deviation. Alternatively, one can search for its maximum and the deviations giving a decrease of 1/2 for its logarithm; in this example, there is no significant difference between the two methods. Note that the estimated error on

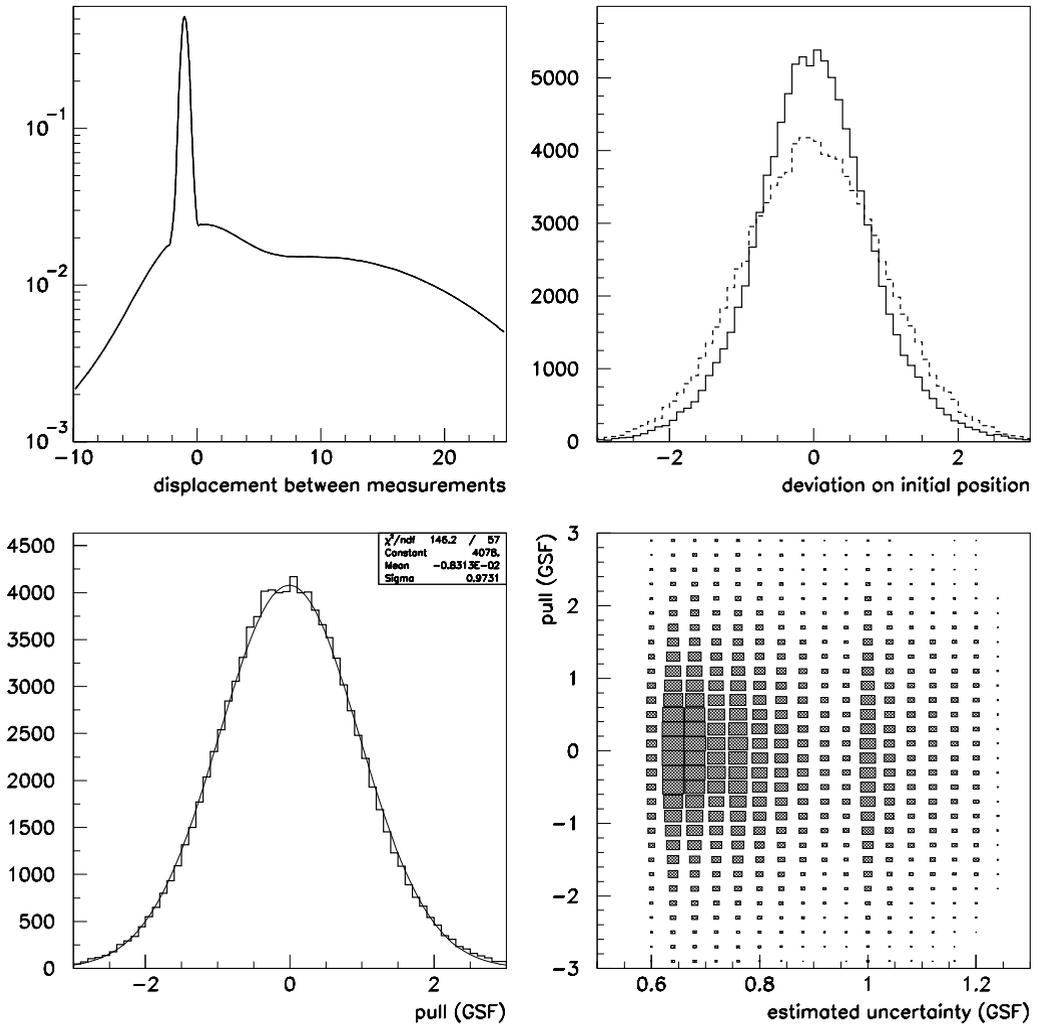


Figure 10. Standard Filter vs Gaussian Sum Filter(GSF) in the simple 1D model (with measurement error = 1. Top left: the distribution of the random displacements between the measurements. Top right: error of the GSF estimator (solid) vs the Standard one. Bottom left: the distribution of pulls for the GSF (global). Bottom right: the pulls for GSF vs the estimated uncertainty.

the gaussian sum depends on the actual configuration of the displacements, while the standard filter gives always the same value.

In Fig.10 we summarize the results on this example: the gaussian sum gives a slightly narrower distribution of errors. More important, it provides an estimation of the variance for each realization, which extends over a large range: computing the “pulls” (deviation/error) with this estimation gives the right spread for any value of the variance. In brief, the average error is not greatly improved,

but we have a distinction between more or less precise evaluations, with a reliable error for every configuration.

5.3 Comments

The main application of a fitting procedure extended beyond the gaussian approximation is the reconstruction of electrons/positrons, accounting for the tail in the distribution of energy loss. We can try to understand intuitively what can be gained. The trajectory is measured over a given segment: if a large energy loss occurs close to the end of this segment, it has no significant effect, whatever the procedure; if it occurs close to the beginning, both the gaussian and the beyond-to-gaussian methods will suffer the same bias on the energy. There may be a significant difference if the large loss occurs in the central region of the segment: the beyond-to-gaussian backward filter includes a tail towards lower curvature (larger energy), so it has more flexibility to modify the curvature when including the points before the large loss, and hence to obtain a better evaluation of the initial energy.

In principle, the formalism may be used outside the measurement range, for example, when including a calorimetric measurement: it can be transformed through an extrapolation to the trajectometer, accounting for the material in between, to give a non-gaussian distribution for the curvature at the beginning of the backward filter (even if was roughly gaussian within the calorimeter). Similarly, the backward extrapolation to the vertex region may be beyond-to-gaussian, including the material crossed before the trajectometer. The non-gaussian features can be introduced in subsequent kinematical reconstructions; in practice, this may be difficult to implement, especially if the trajectometry and the kinematics are handled in independent modules, in the spirit of “hidden boxes” in an Object Oriented framework.

The vertex procedure (Sect.6) may be coupled to the track fitting to improve the reconstruction. For example, if an electron/positron is supposed to come from a given vertex, the position of this vertex can be used as a “virtual measurement” constraining the initial part of the trajectory and improving the reconstruction of the energy. But this is not possible if one wants to decide whether this electron/positron comes from the main interaction or from a secondary decay; in any case, such a decision is more ambiguous than for a heavy particle.

6 Fitting a vertex

Once the trajectories have fitted, we have for each one a 5-vector of parameters \mathbf{p}_i (intersection with the initial surface) with their weight matrix W_i . Assuming that a given sample of trajectories comes from the same vertex of interaction, we want to reconstruct this vertex (and possibly check this assumption). This way be done at two levels:

- find the best estimator of the 3 coordinates of the vertex (and evaluate errors on them). This may also provide a criterion of quality, e.g. a χ^2 providing a probability for the hypothesis of convergence; if possible, we also want to define a criterion for each individual particle to belong to the vertex. Hereafter we call “simple vertex fit” such a procedure.
- exploit the fact that the trajectories come from this point to improve their reconstruction, that is, add to each trajectory a virtual measurement given by the other ones. This is interesting in view of the kinematical reconstruction of the event. All trajectories are improved, and particularly those measured over a short range, because their parameters (especially the curvature) are poorly defined: an additional point may give a very useful information; but, of course, the criterion to decide whether such a trajectory should be attached to the common vertex is loose. This “full vertex fit” is *a priori* a complex procedure, because we want to fit $3N + 3$ parameters: the coordinates of the vertex and 3

quantities to define the initial state of N particles (e.g. p_x, p_y, p_z or better $1/p$ and two angles). We will see how the problem can be simplified using a linearization.

6.1 The simple vertex fit

The procedure is conceptually simple. From the initial parameters one can deduce the parameters and their error matrix on any surface: this defines a “tube” of probability around the trajectory. When extrapolating the trajectory backwards from the initial surface to the region of the vertex, the errors on the position increase, so the tube gets broader, but over a short range it may be considered as a cylinder, as illustrated in Fig.11. In other terms, if the position of the vertex is approximately known, each trajectory provides an information on the vertex coordinates that may be summarized in a position with a weight matrix of rank 2 (the position may be arbitrary chosen along the axis of the tube): as in Sect.3, combining these informations amounts to make their weighted mean. If at least two non parallel tubes are combined, the degeneracy of the position is removed.

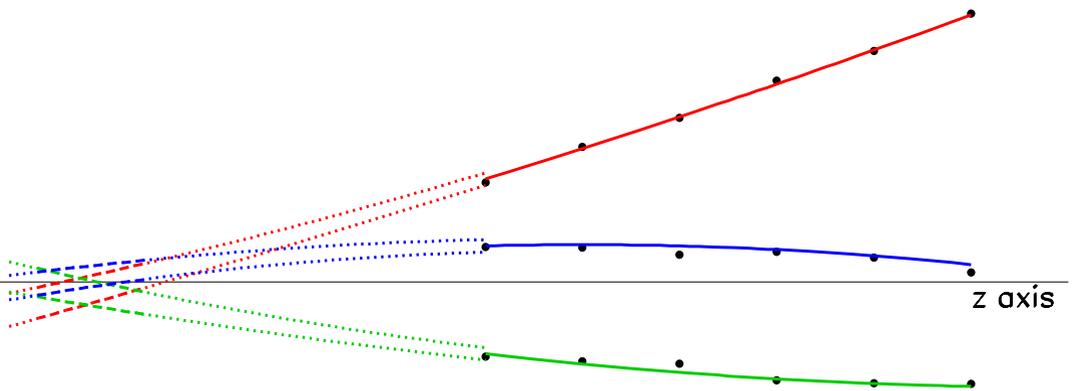


Figure 11. Principle of the “simple” vertex fit. Each trajectory fitted to measurements (black points) provides an initial position with its 2×2 error matrix, which is extrapolated backwards to the vertex region (dotted lines). The errors increase with the range of the extrapolation, but in the region of interest (dashed lines) they can be represented by a cylinder, that is, an arbitrary position along a straight line and a constant error matrix on two coordinates (e.g. here x, y at fixed z). Finding the vertex consists in making the *weighted mean* of these tubes, in the sense defined in Sect.3.1.

We describe now the mathematical procedure. Let us suppose that the parameters are x, y at fixed z , and three more for the direction and the curvature. First, an approximate position (x_0, y_0, z_0) of the vertex is found from the intersections of the extrapolated trajectories in xz and yz projections. Then, the error matrix C_i of each trajectory is propagated to z_0 , using the matrix of derivatives $\mathcal{D}_i = \partial \mathbf{p}_0 / \partial \mathbf{p}_i$; actually, we just need the 2×5 submatrix D_i of the derivatives of x, y (at fixed $z = z_0$) w.r.t. \mathbf{p}_i to compute the 2×2 error matrix $C_{i0} = D_i C_i D_i^T$ and $W_{i0} = C_{i0}^{-1}$. If we approximate locally the trajectory as $x = x_0 + a_x(z - z_0)$, $y = y_0 + a_y(z - z_0)$, we can describe the probability of presence of the vertex at $\mathbf{v} = (x, y, z)$ by saying that the 2-vector $(x - a_x(z - z_0), y - a_y(z - z_0)) = \mathbf{u} - (z - z_0)\mathbf{a}$ has a mean position \mathbf{u}_0 with a weight matrix W_{i0} . In the gaussian approximation, this gives a density of

probability $\exp(-W_{i0} [\mathbf{u} - (z - z_0)\mathbf{a} - \mathbf{u}_0] / 2)$, that is a 3×3 weight matrix for \mathbf{v} (writing W for W_{i0} in the right hand side):

$$W_{i0} = \begin{pmatrix} W_{xx} & W_{xy} & -a_x W_{xx} - a_y W_{xy} \\ & W_{yy} & -a_x W_{xy} - a_y W_{yy} \\ & & a_x^2 W_{xx} + 2a_x a_y W_{xy} + a_y^2 W_{yy} \end{pmatrix}$$

Let $\mathbf{v}_{i0} = (\mathbf{u}_{i0}, z_0)$ be the mean position of the extrapolation of the trajectory i . Its weight matrix W_{i0} has rank 2, but if the extrapolations of trajectories at z_0 are not all parallel, the weighted mean of the extrapolated positions of the trajectories is defined as $(\sum_i W_{i0})^{-1} \sum_i W_{i0} \mathbf{v}_{i0}$: this is an estimator of the vertex position (the best one in the gaussian case), with an error matrix $(\sum_i W_{i0})^{-1}$.

6.2 The full vertex fit as a "hierarchical" fit

This procedure ([3, 7, 9]) aims to fit $3N + 3$ parameters ($\mathbf{V} = (x_v, y_v, z_v)$ and $\mathbf{q}_i = (1/p_i, \theta_{vi}, \varphi_{vi})$ for each of the N particles) to a set of N 5-fold measurements, e.g. $\mathbf{p}_k = (x_i, y_i, 1/p_i, \theta_i, \varphi_i)$ at a fixed value of z for particle i , with a weight matrix W_k . The equation of propagation expresses \mathbf{p}_k as a function of V and \mathbf{q}_k , and we see that the parameters to be fitted do not play the same role: for any k , \mathbf{p}_i depends on \mathbf{V} , but not on \mathbf{q}_j if $j \neq i$. So we can distinguish 3 *global* parameters and N *individual* subsets of 3 parameters, which are related to only one measurement: we call "hierarchical" such a fit.

If we want to perform the fit by minimizing a function F (χ^2 or negative log-likelihood), we can write it as a sum over the particles (measured independently):

$$F(\mathbf{V}, \mathbf{q}_1, \dots, \mathbf{q}_N) = \sum_i f(\mathbf{p}_i(\mathbf{V}, \mathbf{q}_i)) \quad \rightarrow \quad \min F = \min_{\mathbf{V}} \sum_i \min_{\mathbf{p}_i} f(\mathbf{p}_i(\mathbf{V}, \mathbf{q}_i))$$

So, if we want to use a minimizing package, we can use an embedded structure of minimizers, where the function of $3N + 3$ parameters to be minimized is itself computed at each step through N calls to a minimizer of a function of 3 parameters. In this procedure, the correlations between all parameters are taken into account to find the best path towards the minimum.

We can imagine another solution, using an iterative alternate procedure: fit V with fixed \mathbf{q}_k , then fit each \mathbf{q}_k with fixed V , and so on. In practice, if the parameters are correlated, the convergence may be very slow⁵. It may be accelerated if F is quasi quadratic around its minimum: in this favourable case, the differences between the parameters at step i and their final values decrease exponentially with i , so after a certain number of steps, one can evaluate a good approximation of the limits, restart the alternate fit, redo the computation of limits, and restart the overall procedure if needed. An advantage of this method is to offer a better control of the convergence, and a way to remove tracks during the iteration, while the first one uses the minimizer package as a "black box" where the internal strategy cannot be modified.

Whatever the strategy, the computing time grows rapidly with N . If the particles produced in the initial interaction may produce secondary vertices (decays or interactions in the material), one has to make trials to choose to best association of tracks to vertices, and the computation may become very heavy. Fortunately, in most detectors, the \mathbf{p}_k depend linearly on the variations of \mathbf{V} and \mathbf{q}_k within a few standard deviations around the central values: with this approximation, minimizing a global χ^2 depending on $3N + 3$ parameters will result in a sparse linear system of equations, that can be solved

⁵ a similar situation occurs when going to the minimum of $g(x, y)$ by alternate downgoing steps at fixed x and y : if the valley has elliptical contours with oblique axes, it is easy to see that the convergence is slow when the ellipses have a large length/width ratio.

through a number of operations proportional to N , and moreover adding a removing a track to/from a vertex will be simple.

6.3 Linearization of the problem

Let \mathbf{V}_0 be an approximate position of the vertex, and \mathbf{q}_{i0} approximate parameters of track i at the vertex (e.g. the ones obtained by a backward extrapolation of \mathbf{p}_i to z_0). In the linear approximation we differentiate the propagation from \mathbf{V}, \mathbf{q}_i to \mathbf{p}_i :

$$\mathbf{p}_i(\mathbf{V}_0 + \delta\mathbf{V}, \mathbf{q}_{i0} + \delta\mathbf{q}_i) = \mathbf{p}_i(\mathbf{V}_0, \mathbf{q}_{i0}) + D_i \delta\mathbf{V} + E_i \delta\mathbf{q}_i$$

where D_i and E_i are (5×3) matrices of derivatives⁶. So, if each track was fitted individually as (\mathbf{p}_i^f, W_i) at the initial point, we can rewrite $\chi^2 = \sum_i W_i [\mathbf{p}_i - \mathbf{p}_i^f]^2$ as:

$$\chi^2 = \sum_i W_i [\mathbf{p}_i(\mathbf{V}_0, \mathbf{q}_{i0}) - \mathbf{p}_i^f + D_i \delta\mathbf{V} + E_i \delta\mathbf{q}_i]^2 = \sum_i W_i [\Delta\mathbf{p}_i - D_i \delta\mathbf{V} - E_i \delta\mathbf{q}_i]^2$$

where we have introduced the 5-vector of deviations of the individual fits from the predictions of the first approximation at the vertex: $\Delta\mathbf{p}_i = \mathbf{p}_i^f - \mathbf{p}_i(\mathbf{V}_0, \mathbf{q}_{i0})$.

Using the image of “tubes” in the space of parameters, we can say that the fit of the trajectory i defines a tube of rank 5 in the 6D space $(\mathbf{V}, \mathbf{q}_i)$, and we have to combine N such tubes in a $(3N + 3)$ -D space. The χ^2 is here approximated by a quadratic function of the parameters, so the minimum is given by a linear system of $3N + 3$ equations on $\delta\mathbf{V}$ and the $\delta\mathbf{q}_i$. This system may be split in $N + 1$ blocks of 3 equations; the first block contains terms for all parameters:

$$\sum_i (D_i^T W_i D_i) \delta\mathbf{V} + \sum_i (D_i^T W_i E_i) \delta\mathbf{q}_i = \sum_i (D_i^T W_i) \Delta\mathbf{p}_i \quad \text{in short} \quad \mathcal{A} \delta\mathbf{V} + \sum_i \mathcal{B}_i \delta\mathbf{q}_i = \mathcal{U}$$

the N following ones contain each $\delta\mathbf{V}$ and only one of the $\delta\mathbf{q}_i$:

$$(E_i^T W_i D_i) \delta\mathbf{V} + (E_i^T W_i E_i) \delta\mathbf{q}_i = (E_i^T W_i) \Delta\mathbf{p}_i \quad \text{in short} \quad \mathcal{B}_i^T \delta\mathbf{V} + C_i \delta\mathbf{q}_i = \mathcal{T}_i$$

The last blocks of equations give expressions of the $\delta\mathbf{q}_i$ as a function of $\delta\mathbf{V}$:

$$\delta\mathbf{q}_i = C_i^{-1} (\mathcal{T}_i - \mathcal{B}_i^T \delta\mathbf{V})$$

which can be injected into the first one to obtain an equation giving $\delta\mathbf{V}$:

$$\mathcal{A} \delta\mathbf{V} + \sum_i \mathcal{B}_i C_i^{-1} (\mathcal{T}_i - \mathcal{B}_i^T \delta\mathbf{V}) = \mathcal{U} \quad \rightarrow \quad \delta\mathbf{V} = \left(\mathcal{A} - \sum_i \mathcal{B}_i C_i^{-1} \mathcal{B}_i^T \right)^{-1} \left(\mathcal{U} - \sum_i \mathcal{B}_i C_i^{-1} \mathcal{T}_i \right)$$

Then each $\delta\mathbf{q}_i$ follows from $\delta\mathbf{V}$.

The left hand side of the linear system written above may be expressed with a sparse matrix \mathcal{W} of 3×3 blocks, where only the first line, the first row and the diagonal are non-zero, easily solved by a substitution method:

$$\begin{pmatrix} \mathcal{A} & \cdots & \mathcal{B}_i^T & \cdots \\ \vdots & \ddots & & \\ \mathcal{B}_i & & C_i & \\ \vdots & & & \ddots \end{pmatrix} \begin{pmatrix} \delta\mathbf{V} \\ \vdots \\ \delta\mathbf{q}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathcal{U} \\ \vdots \\ \mathcal{T}_i \\ \vdots \end{pmatrix}$$

⁶ We use here the same notation D_i as in Sect.6.1, but the definition and the meaning are different. Note that here D_i goes from vertex to the initial point of the trajectory.

The global covariance matrix on the $3N + 3$ parameters is the inverse of the global weight matrix \mathcal{W} , that is, written by 3×3 blocks:

$$\begin{aligned} \text{cov}(\mathbf{V}, \mathbf{V}) &= \left(\mathcal{A} - \sum_i \mathcal{B}_i \mathcal{C}_i^{-1} \mathcal{B}_i^T \right)^{-1} \\ \text{cov}(\mathbf{V}, \mathbf{q}_i) &= -\text{cov}(\mathbf{V}, \mathbf{V}) \mathcal{B}_i \mathcal{C}_i^{-1} \\ \text{cov}(\mathbf{q}_i, \mathbf{q}_j) &= \delta_{ij} \mathcal{C}_i^{-1} + \mathcal{C}_i^{-1} \mathcal{B}_i^T \text{cov}(\mathbf{V}, \mathbf{V}) \mathcal{B}_j \mathcal{C}_j^{-1} \end{aligned}$$

This provides an error matrix on the position of the vertex and re-evaluated error matrices on the individual particles. In addition, it should be noted that the procedure introduces a correlation between the different particles: this means that, in principle, this correlation should be taken into account when evaluating the uncertainty on physical quantities built with several particles of a vertex, like the total momentum and energy, equivalent masses, etc.

The system has a unique solution for $N > 1$, provided the trajectories are not all parallel at the vertex. The total number of operations needed to solve the system grows proportionally to N if N is large. In the gaussian approximation, the minimum of χ^2 , as found with the solution of the linear system, follows a law of χ^2 with $5N - (3N + 3) = 2N - 3$ degrees of freedom. The associated probability gives a criterion to decide whether the set of N tracks are actually compatible with the hypothesis of a common origin.

6.4 Flexibility and iterative procedures

In many cases, the bundling of tracks in vertices is ambiguous, either because there are short lived particles giving a secondary vertex close to the main one, or because some secondary particles from remote decays (e.g. K^0 or Λ) are not obviously distinguished. In these conditions, it may be useful to add or remove a track from a vertex to perform a new trial. Let us suppose that we have fitted a vertex with N tracks, giving χ_{\min}^2 . We take the fitted parameters as new starting values; then the χ^2 including a $(N + 1)^{\text{th}}$ track may be written as:

$$\chi^2 = \chi_{\min}^2 + \sum_{i=1}^N W_i [\Delta D_i \delta \mathbf{V} + E_i \delta \mathbf{q}_i] + W_{N+1} [\Delta \mathbf{p}_{N+1} - D_{N+1} \delta \mathbf{V} - E_{N+1} \delta \mathbf{q}_{N+1}]$$

Using \mathcal{A} , \mathcal{B}_i , \mathcal{C}_i , \mathcal{T}_i , \mathcal{U} as defined above, and introducing $\mathcal{A}_{N+1} = D_{N+1}^T W_{N+1} D_{N+1}$ and $\mathcal{U}_{N+1} = D_{N+1}^T W_{N+1} \Delta \mathbf{p}_{N+1}$ the minimization gives:

$$\begin{aligned} (\mathcal{A} + \mathcal{A}_{N+1}) \delta \mathbf{V} + \sum_{i=1}^{N+1} \mathcal{B}_i \delta \mathbf{q}_i &= \mathcal{U}_{N+1} \\ \text{for } i = 1 \text{ to } N \quad \mathcal{B}_i^T \delta \mathbf{V} + \mathcal{C}_i \delta \mathbf{q}_i &= 0 \quad \rightarrow \quad \delta \mathbf{q}_i = -\mathcal{C}_i^{-1} \mathcal{B}_i^T \delta \mathbf{V} \\ \mathcal{B}_{N+1}^T \delta \mathbf{V} + \mathcal{C}_{N+1} \delta \mathbf{q}_{N+1} &= \mathcal{T}_{N+1} \quad \rightarrow \quad \delta \mathbf{q}_{N+1} = \mathcal{C}_{N+1}^{-1} (\mathcal{T}_{N+1} - \mathcal{B}_{N+1}^T \delta \mathbf{V}) \end{aligned}$$

Injecting in the first equation the expressions of $\delta \mathbf{q}_i$ found in the second and in the third one, we obtain an equation which gives $\delta \mathbf{V}$:

$$\left(\mathcal{A} + \mathcal{A}_{N+1} - \sum_{i=1}^{N+1} \mathcal{B}_i \mathcal{C}_i^{-1} \mathcal{B}_i^T \right) \delta \mathbf{V} = \mathcal{U}_{N+1} - \mathcal{B}_{N+1} \mathcal{C}_{N+1}^{-1} \mathcal{T}_{N+1}$$

hence the $\delta\mathbf{q}_i$. The matrices $(\mathcal{A} - \sum_{i=1}^{N+1} \mathcal{B}_i C_i^{-1} \mathcal{B}_i^T)$ and $C_i^{-1} \mathcal{B}_i^T$ were already computed in the fit with N tracks, so the amount of computation needed is much less than doing a fit with $N + 1$ tracks *ab initio*.

The same algorithm can be applied to remove a track from a vertex fit: it is equivalent to add this track with a negative weight matrix $-W_i$. Note also that all operations (multiplications, inversions) involve always 3×5 and 3×3 matrices and may be coded explicitly in a very efficient way, avoiding any use of indexed arrays.

The tools described above can be inserted in an iterative vertex building. In most detectors the trajectories around the vertex may be described by smooth functions, so the linear approximation is quite adequate, especially when using the “perigee” parameters (no iteration is needed for a given set of tracks). With quality criteria based for example on the probability of χ^2 , a strategy may be defined to determine the best repartition of the tracks in vertices; this strategy may be driven by physical considerations, for example finding the decay of a heavy flavour particle, starting from a “seed” (a large p_t lepton, a combination identified by equivalent masses, etc).

6.5 Fitting a vertex with constraints

6.5.1 Beam profile

The simplest case of constraint for the main vertex is the beam profile, which may be considered as a particular trajectory entering the vertex, except that its parameters should not be re-evaluated in the procedure. If the z axis is chosen along the beam, the lateral profile is usually summarized by the lateral standard deviations σ_x and σ_y , and the constraint may be expressed by just adding a term $(x_v/\sigma_x)^2 + (y_v/\sigma_y)^2$ to the χ^2 . In the “simple vertex” procedure (Sect.6.1), we just need to add a trajectory with $x_0 = y_0 = a_x = a_y = 0$ and a diagonal weight matrix $(1/\sigma_x^2, 1/\sigma_y^2)$. In the “full vertex” fit within the linear approximation (Sect.6.3), the additional term results in adding the diagonal matrix $W_b = (1/\sigma_x^2, 1/\sigma_y^2, 0)$ to \mathcal{A} and $-W_b \mathbf{V}_0$ to \mathcal{U} .

This formalism may be applied to both fixed target experiments and colliders, but it improves the results only if σ_x and σ_y are comparable to, or smaller than the errors on extrapolated tracks: in practice, this is true in colliders, where the beam constraint is very useful.

6.5.2 Kinematical and geometrical constraints

A typical example is the reconstruction of the so called V^0 's: remote decay of a neutral object (γ, K^0, Λ , etc) in two charged particles. The constraint consists in the value of the equivalent mass of the pair with given mass assumptions for the charged particles i and j . The fit may be performed by removing one of the free parameters and replacing it by a function of the other ones; in the case of $\gamma \rightarrow e^+ e^-$, we can force \vec{p}_i and \vec{p}_j to be parallel by making a fit with 7 parameters instead of 9: x_v, y_v, z_v , a common value for θ and φ , and the curvatures c_i and c_j . More generally, the constraint may be expressed as $F(\mathbf{p}_i, \mathbf{p}_j) = 0$ and handled through the generic method of Lagrange multipliers (see below).

As a result of the fit, we obtain estimators for the trajectory of the neutral particle (a straight line), which can be inserted in a primary or a nearby secondary vertex, through one of the procedures described above: the introduction of 4-vectors instead of 5-vectors in the formalism is straightforward. Conversely, we can introduce an additional constraint in the remote vertex fit, if the neutral particle comes from a previously fitted vertex with a given error matrix.

6.5.3 General procedure with Lagrange multipliers in the linear approximation

The constraint may be applied as a further step after the "standard" fit (with the pure constraint of convergence). Let us call now \mathbf{p}^f the global vector of values fitted without the constraint $F(\mathbf{p}) = 0$. We can suppose that the constraint is nearly satisfied by \mathbf{p}^f , that is, F can be linearly expanded in the neighbourhood of \mathbf{p}^f :

$$F(\mathbf{p}) = F(\mathbf{p}^f) + \nabla F \cdot (\mathbf{p} - \mathbf{p}^f)$$

where ∇F is the gradient of F at \mathbf{p}^f . On the other hand, the χ^2 may be approximated by a quadratic expansion around \mathbf{p}^f :

$$\chi^2 = \chi_{\min}^2 + \mathcal{W} [\mathbf{p} - \mathbf{p}^f]$$

The Lagrange multiplier method consists in finding the minimum of:

$$G(\mathbf{p}) = \frac{1}{2} (\chi_{\min}^2 + \mathcal{W} [\mathbf{p} - \mathbf{p}^f]) + \lambda (F(\mathbf{p}^f) + \nabla F \cdot (\mathbf{p} - \mathbf{p}^f))$$

when varying both the components of \mathbf{p} and λ . This is a quadratic function, so the solution can be obtained easily in two steps:

- canceling the gradient of G w.r.t. \mathbf{p} , which provides \mathbf{p} as a linear function of λ :

$$\nabla G(\mathbf{p}) = \mathcal{W} [\mathbf{p} - \mathbf{p}^f] + \lambda \nabla F = 0 \quad \rightarrow \quad \mathbf{p} - \mathbf{p}^f = -\lambda \mathcal{W}^{-1} \cdot \nabla F$$

- canceling the derivative w.r.t. λ and introducing the previous expression to solve a linear equation in λ :

$$\frac{\partial G}{\partial \lambda} = F(\mathbf{p}^f) + \nabla F \cdot (\mathbf{p} - \mathbf{p}^f) = F(\mathbf{p}^f) + \lambda \mathcal{W}^{-1} [\nabla F] \quad \rightarrow \quad \lambda = \frac{F(\mathbf{p}^f)}{\mathcal{W}^{-1} [\nabla F]}$$

This value of λ gives \mathbf{p} through the first equation.

This procedure is easily extended to the case of several simultaneous constraints $F_k(\mathbf{p})$, $k = 1 \cdots N_c$:

$$G(\mathbf{p}) = \frac{1}{2} (\chi_{\min}^2 + \mathcal{W} [\mathbf{p} - \mathbf{p}^f]) + \sum_k \lambda_k (F(\mathbf{p}^f) + \nabla F_{k \cdot} (\mathbf{p} - \mathbf{p}^f))$$

We obtain $\mathbf{p} - \mathbf{p}^f$ as a linear function of the λ_k and a linear system of equations on the λ_k , where the expressions of $\mathbf{p} - \mathbf{p}^f$ may be injected to eliminate them. The values of the λ_k obtained from this linear system provide the wanted solution for \mathbf{p} .

7 Appendix: derivative matrix for propagation in the helix model

With the notations defined in Sect.3.5, the trajectory is defined through a running parameter φ :

$$\begin{aligned} x &= x_0 + \mathcal{R} (\sin \varphi - \sin \varphi_0) = r_0 \cos \Phi_0 + \mathcal{R} (\sin \varphi - \sin \varphi_0) \\ y &= y_0 - \mathcal{R} (\cos \varphi - \cos \varphi_0) = r_0 \sin \Phi_0 + \mathcal{R} (\cos \varphi - \cos \varphi_0) \\ z &= z_0 + \mathcal{R} \cot \theta (\varphi - \varphi_0) \end{aligned}$$

We define the signed curvature $c = 1/\mathcal{R}$. To simplify expressions we use $t = \cot \theta$ as parameter instead of θ . We want to obtain analytical expression for the derivatives of the non-constant parameters (i.e. other than t, c) on a surface, as functions of the parameters on another one.

7.1 Propagation between planes

7.1.1 Planes perpendicular to the magnetic field

The initial parameters are x, y, φ, t, c at fixed z_0 . To obtain the parameters at $z = z_1$, we can write, with $\Delta z = z_1 - z_0$ and $\ell = \mathcal{R} \Delta \varphi = \Delta z/t$ (length of the arc in xy projection):

$$\varphi_1 = \varphi_0 + \frac{c \Delta z}{t}$$

hence the derivatives:

$$\frac{\partial \varphi_1}{\partial \varphi_0} = 1 \quad ; \quad \frac{\partial \varphi_1}{\partial t} = -\frac{c \Delta z}{t^2} = -\frac{c \ell}{t} \quad ; \quad \frac{\partial \varphi_1}{\partial c} = \frac{\Delta z}{t} = \ell$$

The expressions for x_1, y_1 and their derivatives follow immediately from φ_1 , using the notation $\Delta A = A_1 - A_0$ for any quantity A depending on the local parameters:

$$\begin{aligned} \frac{\partial x_1}{\partial x_0} &= \frac{\partial y_1}{\partial y_0} = 1 \quad ; \quad \frac{\partial x_1}{\partial y_0} = \frac{\partial x_1}{\partial t} = \frac{\partial y_1}{\partial x_0} = \frac{\partial y_1}{\partial t} = 0 \\ \frac{\partial x_1}{\partial \varphi_0} &= \mathcal{R} \left(-\cos \varphi_0 + \cos \varphi_1 \frac{\partial \varphi_1}{\partial \varphi_0} \right) = \frac{\Delta(\cos \varphi)}{c} \quad ; \quad \frac{\partial x_1}{\partial c} = -\frac{\Delta(\sin \varphi)}{c^2} \\ \frac{\partial y_1}{\partial \varphi_0} &= \mathcal{R} \left(-\sin \varphi_0 + \sin \varphi_1 \frac{\partial \varphi_1}{\partial \varphi_0} \right) = \frac{\Delta(\sin \varphi)}{c} \quad ; \quad \frac{\partial y_1}{\partial c} = \frac{\Delta(\cos \varphi)}{c^2} \end{aligned}$$

In the approximation of weak curvature ($|\Delta \varphi| \ll 1$) we obtain “quasi straight line” approximations for the last two:

$$\frac{\partial x_1}{\partial \varphi_0} \simeq -\ell \sin \varphi_0 \quad ; \quad \frac{\partial x_1}{\partial c} \simeq -\frac{\ell \cos \varphi_0}{c} \quad ; \quad \frac{\partial y_1}{\partial \varphi_0} \simeq \ell \cos \varphi_0 \quad ; \quad \frac{\partial y_1}{\partial c} \simeq -\frac{\ell \sin \varphi_0}{c}$$

7.1.2 Planes parallel to the magnetic field

The parameters are now y, z, φ, t, c at fixed x ; we want to go from x_0 to x_1 . As previously, we first determine φ_1 and its derivatives, using similar notations (assuming that the right solution φ_1 of the trigonometric equation is chosen; usually it is the closest one to φ_0); some of the computations are the same as in Sect. 3.4.

$$\sin \varphi_1 = \sin \varphi_0 + c \Delta x \quad \rightarrow \quad \frac{\partial \varphi_1}{\partial \varphi_0} = \frac{\cos \varphi_0}{\cos \varphi_1} \quad , \quad \frac{\partial \varphi_1}{\partial c} = \frac{\cos \varphi_0}{\cos \varphi_1} \Delta x$$

(the other derivatives vanish)

Hence the derivatives of the other parameters which are not 1 or 0:

$$\begin{aligned} \frac{\partial y_1}{\partial \varphi_0} &= \frac{\sin(\Delta \varphi)}{c \cos \varphi_1} \simeq \frac{\ell}{c \cos \varphi_0} \quad ; \quad \frac{\partial y_1}{\partial c} = \frac{1 - \cos(\Delta \varphi)}{c^2 \cos \varphi_1} \simeq \frac{\ell^2}{2 \cos \varphi_1} \\ \frac{\partial z_1}{\partial \varphi_0} &= \frac{t \Delta(\cos \varphi)}{c \cos \varphi_1} \simeq -t \tan \varphi_0 \ell \quad ; \quad \frac{\partial z_1}{\partial t} = \frac{\Delta \varphi}{c} = \ell \quad ; \quad \frac{\partial z_1}{\partial c} = -\frac{t \Delta \varphi}{c^2} \simeq -\frac{t \ell}{c} \end{aligned}$$

7.2 Propagation between cylinders

In the following, we will use as trajectory parameters at fixed r : Φ , ξ and \mathcal{R} (or $c = 1/\mathcal{R}$) in xy projection, and z , $t = \cot \theta$ to complete the description in 3D space. The coordinates of the center C of the projection onto the xy plane may be expressed from any point (r, Φ, φ) on the trajectory as:

$$x_c = r \cos \Phi - \mathcal{R} \sin \varphi \quad (1)$$

$$y_c = r \sin \Phi + \mathcal{R} \cos \varphi \quad (2)$$

so the polar coordinates of C are r_c , Φ_c such that:

$$r_c^2 = r^2 + \mathcal{R}^2 - 2\mathcal{R}r \sin \xi \quad \Phi_c = \text{atan2}(y_c, x_c)$$

where we define $\xi = \varphi - \Phi$ (deviation from the radial direction).

As intermediate parameters, we use also r_c defined above and the z coordinate of the perigee $z_p = z - \mathcal{R}t(\varphi - \varphi_p)$. Note that with our convention on the geometrical sign S of the curvature (the sign of \mathcal{R}), we have $\Phi_c = \varphi_p + S\pi/2$. To simplify some expressions we introduce for any point on the trajectory $\psi = \varphi - \varphi_p$ (rotation from the perigee to this point). Introducing $\rho_c = S r_c$, we can rewrite (1) and (2) as:

$$-\rho_c \sin \varphi_p = r \cos \Phi - \mathcal{R} \sin \varphi \quad (3)$$

$$\rho_c \cos \varphi_p = r \sin \Phi + \mathcal{R} \cos \varphi \quad (4)$$

Then the combinations $\cos \varphi(3) + \sin \varphi(4)$ and $\cos \varphi(4) - \sin \varphi(3)$ give:

$$\cos \psi = \frac{\mathcal{R} - r \sin \xi}{\rho_c} \quad ; \quad \sin \psi = \frac{r \cos \xi}{\rho_c}$$

The notations used in this Section are illustrated in Fig. 12.

We want to obtain the derivatives of the parameters at $r = r_1$ with respect to the parameters at $r = r_0$: to do so we will compute the derivatives of the intermediate parameters (r_c, Φ_c, z_p) with respect to the initial ones and to the final ones, and use the inversion and the multiplication of jacobian matrices. For convenience, we compute derivatives w.r.t. \mathcal{R} instead of c in the intermediate steps. First we consider the transformation from $(\Phi_0, \xi_0, \mathcal{R}, z_0, t)$ to $(\rho_c, \Phi_c, \mathcal{R}, z_p, t)$. From the expression of $\rho_c^2 = r_c^2 = r_0^2 + \mathcal{R}^2 - 2\mathcal{R}r_0 \sin \xi_0$, we have immediately:

$$\begin{aligned} \frac{\partial \rho_c}{\partial \Phi_0} &= \frac{\partial \rho_c}{\partial z_0} = \frac{\partial \rho_c}{\partial t} = 0 \\ \frac{\partial \rho_c}{\partial \xi_0} &= -\frac{\mathcal{R}r_0 \cos \xi_0}{\rho_c} = -\mathcal{R} \sin \psi_0 \quad ; \quad \frac{\partial \rho_c}{\partial \mathcal{R}} = -\frac{r_0 \sin \xi_0}{\rho_c} = \cos \psi_0 - \frac{\mathcal{R}}{\rho_c} \end{aligned}$$

For the derivatives of Φ_c , we use $d(\text{atan2}(y_c, x_c)) = (x_c dy_c - y_c dx_c)/(x_c^2 + y_c^2)$, hence for any parameter α : $\partial \Phi_c / \partial \alpha = (x_c \partial y_c / \partial \alpha - y_c \partial x_c / \partial \alpha) / \rho_c^2$. This gives:

$$\begin{aligned} \frac{\partial \Phi_c}{\partial \Phi_0} &= 1 \quad (\text{obvious by rotational invariance}) \quad ; \quad \frac{\partial \Phi_c}{\partial z_0} = \frac{\partial \Phi_c}{\partial t} = 0 \\ \frac{\partial \Phi_c}{\partial \xi_0} &= \frac{\mathcal{R}(\mathcal{R} - r_0 \sin \xi_0)}{\rho_c^2} = \frac{\mathcal{R} \cos \psi_0}{\rho_c} \quad ; \quad \frac{\partial \Phi_c}{\partial \mathcal{R}} = \frac{r_0 \cos \xi_0}{\rho_c^2} = S \frac{\sin \psi_0}{\rho_c} \end{aligned}$$

The derivatives of z_p are obtained from $z_p = z_0 - \mathcal{R}t\psi_0 = z_0 + \mathcal{R}t(\Phi_c - \Phi - \xi - S\pi/2)$; to shorten one expression, we also introduce $\Psi = \Phi - \Phi_c$, with $\cos \Psi = (r - \mathcal{R} \sin \xi) / r_c$:

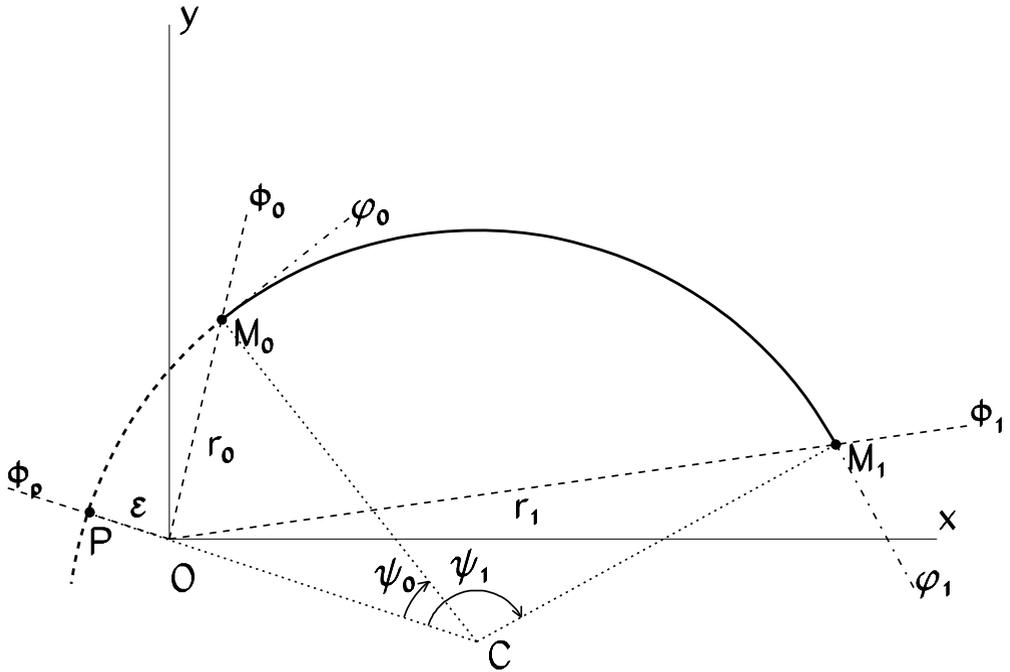


Figure 12. Parameters used to express the propagation of cylindrical parameters at fixed r .

$$\begin{aligned} \frac{\partial z_p}{\partial \Phi_0} &= 0 \quad ; \quad \frac{\partial z_p}{\partial z_0} = 1 \\ \frac{\partial z_p}{\partial \xi_0} &= \mathcal{R} t \left(\frac{\partial \Phi_c}{\partial \xi_0} - 1 \right) = \frac{\mathcal{R} r_0 t (\mathcal{R} \sin \xi_0 - r_0)}{\rho_c^2} = -\frac{\mathcal{R} r_0 t \cos \Psi_0}{r_c} \\ \frac{\partial z_p}{\partial t} &= -\mathcal{R} \psi_0 \\ \frac{\partial z_p}{\partial \mathcal{R}} &= t \left(\psi_0 + \mathcal{R} \frac{\partial \Phi_c}{\partial \mathcal{R}} \right) = t \left(\psi_0 + \frac{\mathcal{R} \sin \psi_0}{\rho_c} \right) \end{aligned}$$

The derivatives form the jacobian matrix

$$D_{0 \rightarrow c} = \frac{\partial(\rho_c, \Phi_c, z_p, t, \mathcal{R})}{\partial(\xi_0, \Phi_0, z_0, t, \mathcal{R})} = \begin{pmatrix} -\mathcal{R} \sin \psi_0 & 0 & 0 & 0 & \cos \psi_0 - \mathcal{R}/\rho_c \\ \mathcal{R} \cos \psi_0 / \rho_c & 1 & 0 & 0 & \sin \psi_0 / \rho_c \\ -\mathcal{R} r_0 t \cos \Psi_0 / r_c & 0 & 1 & -\mathcal{R} \psi_0 & t(\psi_0 + \mathcal{R} \sin \psi_0 / \rho_c) \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrix $D_{1 \rightarrow c} = \partial(\rho_c, \Phi_c, z_p, t, \mathcal{R}) / \partial(\xi_1, \Phi_1, z_1, t, \mathcal{R})$ has exactly the same expression, replacing the index 0 by 1. To obtain matrix $D_{0 \rightarrow 1}$ of the derivatives of $(\xi_1, \Phi_1, z_1, t, \mathcal{R})$ w.r.t. $(\xi_0, \Phi_0, z_0, t, \mathcal{R})$ we just

need to invert $D_{1 \rightarrow c}$ and then:

$$D_{0 \rightarrow 1} = D_{c \rightarrow 1} D_{0 \rightarrow c} = (D_{1 \rightarrow c})^{-1} D_{0 \rightarrow c}$$

Note that the sparse structure of $D_{1 \rightarrow c}$ is conserved by inversion; for any values of (a, b, c, d, e, f, g) we have:

$$\begin{pmatrix} a & 0 & 0 & 0 & e \\ b & 1 & 0 & 0 & f \\ c & 0 & 1 & d & g \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1/a & 0 & 0 & 0 & -e/a \\ -b/a & 1 & 0 & 0 & be/a - f \\ -c/a & 0 & 1 & -d & ce/a - g \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This expression makes the algebraic computations easy:

$$\begin{pmatrix} a_1 & 0 & 0 & 0 & e_1 \\ b_1 & 1 & 0 & 0 & f_1 \\ c_1 & 0 & 1 & d_1 & g_1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} a_0 & 0 & 0 & 0 & e_0 \\ b_0 & 1 & 0 & 0 & f_0 \\ c_0 & 0 & 1 & d_0 & g_0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{a_0}{a_1} & 0 & 0 & 0 & \frac{e_0 - e_1}{a_1} \\ b_0 - b_1 \frac{a_0}{a_1} & 1 & 0 & 0 & f_0 - f_1 + (e_1 - e_0) \frac{b_1}{a_1} \\ c_0 - c_1 \frac{a_0}{a_1} & 0 & 1 & d_0 - d_1 & g_0 - g_1 + (e_1 - e_0) \frac{c_1}{a_1} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

We give the derivatives which are not 1 or 0, using $\Delta\psi = \Delta\varphi$, with some useful approximations as function of $\ell = \mathcal{R}\Delta\varphi$ (length of the arc in xy projection) in the case of low curvature (more precisely: $\Delta\varphi \ll 1$):

$$\begin{aligned} \frac{\partial\Phi_1}{\partial\xi_0} &= -\frac{R \sin(\Delta\varphi)}{r_c \sin\psi_1} = \frac{\mathcal{R} \sin(\Delta\varphi)}{r_1 \cos\xi_1} \simeq \frac{\ell}{r_1 \cos\xi_1} \\ \frac{\partial\Phi_1}{\partial\mathcal{R}} &= \frac{\cos(\Delta\varphi) - 1}{\rho_c \sin\psi_1} = \frac{\cos(\Delta\varphi) - 1}{r_1 \cos\xi_1} \rightarrow \frac{\partial\Phi_1}{\partial c} = \frac{R^2(1 - \cos(\Delta\varphi))}{r_1 \cos\xi_1} \simeq \frac{\ell^2}{2r_1 \cos\xi_1} \\ \frac{\partial\xi_1}{\partial\xi_0} &= \frac{\sin\psi_0}{\sin\psi_1} \simeq 1 \\ \frac{\partial\xi_1}{\partial\mathcal{R}} &= \frac{\Delta(\cos\psi)}{\mathcal{R} \sin\psi_1} \rightarrow \frac{\partial\xi_1}{\partial c} = -\frac{\mathcal{R}\Delta(\cos\psi)}{\sin\psi_1} \simeq \ell \\ \frac{\partial z_1}{\partial\xi_0} &= \frac{\mathcal{R}t}{r_c} \left(\frac{\sin\psi_0}{\sin\psi_1} r_1 \cos\Psi_1 - r_0 \cos\Psi_0 \right) \\ \frac{\partial z_1}{\partial t} &= \mathcal{R}\Delta\psi = \ell \\ \frac{\partial z_1}{\partial c} &= \mathcal{R}^2 t \left(-\Delta\varphi + \frac{\mathcal{R}\Delta(\sin\psi)}{\rho_c} + \frac{r_1}{r_c} \frac{\Delta(\cos\psi) \cos\Psi_1}{\sin\psi_1} \right) \end{aligned}$$

Some approximations may be applied to the first and last derivative of z_1 if the impact parameter $|R - r_c|$ is small.

References

- [1] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME - Journal of Basic Engineering **82** (1960) 35
 R.E. Kalman, R. S. Bucy, "New Results in Linear Filtering and Prediction Theory", Transactions of the ASME - Journal of Basic Engineering **83** (1961) 95

- [2] P. Billoir, “Track fitting with multiple scattering: a new method”, Nucl. Inst. Meth., **225** (1984) 352
- [3] P. Billoir, R. Früwirth, M. Regler, “Track element merging strategy and vertex fitting in complex modular storage ring detectors”, Nucl. Inst. Meth., **241** (1985) 115
- [4] R. Früwirth, “Application of Kalman filtering to track and vertex fitting” Nucl. Inst. Meth., **262** (1987) 444
- [5] P. Billoir, “Progressive track recognition with a Kalman-like fitting procedure”, Comp. Phys. Comm. **57** (1989) 390
- [6] P. Billoir, S.Qian, “Simultaneous pattern recognition and track fitting by the Kalman Filtering method”, Nucl. Inst. Meth. **A294** (1990) 219
- [7] P. Billoir, S.Qian, “Fast vertex fitting with a local parametrization of tracks”, Nucl. Inst. Meth. **A311** (1990) 219
- [8] R. Früwirth, “Track fitting with non-gaussian noise”, Comp. Phys. Comm. **110** (1997) 1
R. Früwirth, S. Früwirth-Schnatter, “On the treatment of energy loss in track fitting”, Comp. Phys. Comm. **110** (1998) 80
- [9] Tools for hierarchical fits in ROOT framework: <http://gentitfx.fr/Hierarchal>