Kalman Filter Track Fits and Track Breakpoint Analysis

Pierre Astier [a] Alessandro Cardini [b] Robert D. Cousins [b,1]
Antoine Letessier-Selvon [a] Boris A. Popov [a,2]
Tatiana Vinogradova [b]

[a] *LPNHE, Laboratoire de Physique Nucléaire et des Hautes Energies, Universités de Paris 6 et 7, 75252 Paris Cedex 05, France.*

[b] *Department of Physics and Astronomy, University of California, Los Angeles, California 90095, U.S.A.*

Abstract

We give an overview of track fitting using the Kalman filter method in the NOMAD detector at CERN, and emphasize how the wealth of by-product information can be used to analyze track breakpoints (discontinuities in track parameters caused by scattering, decay, etc.). After reviewing how this information has been previously exploited by others, we describe extensions which add power to breakpoint detection and characterization. We show how complete fits to the entire track, with breakpoint parameters added, can be easily obtained from the information from unbroken fits. Tests inspired by the Fisher F-test can then be used to judge breakpoints. Signed quantities (such as change in momentum at the breakpoint) can supplement unsigned quantities such as the various chisquares. We illustrate the method with electrons from real data, and with Monte Carlo simulations of pion decays.

Key words: breakpoints, Kalman filter, track fitting
PACS code: 07.05.Kf

1 Introduction

The Kalman filter is an efficient algorithm for fitting tracks in particle spectrometers with many position-sensing detectors [1–6]. It cures many of the problems of traditional χ^2 track fitting using Newton steps, which becomes

[1] Email address: cousins@physics.ucla.edu
[2] on leave from the Laboratory of Nuclear Problems, JINR, 141980 Dubna, Russia.

Accepted for publication in Nuclear Instruments and Methods 14 December 1999

more and more unwieldy as the number of position measurements increases. In such situations where Kalman filtering is naturally applied, it can be possible to detect and characterize track *breakpoints*, defined as locations where one or more of the track parameters is discontinuous. Obvious breakpoints, such as a large kink due to a particle decay, are often found before a full track fit is performed. More subtle breakpoints may only manifest themselves when the track is fit to obtain the track parameters.

Frühwirth[3] has investigated the detection of breakpoints using information which is a natural by-product of a Kalman filter track fit. When fitting a drift-chamber track with N position measurements ("hits"), the idea has been the following. At the location of hit k, one has the best-fit track parameters (a) using hits 1 through k and (b) using hits $k+1$ through N. One constructs a χ^2 for the consistency of these two sets of track parameters. This χ^2, along with other χ^2's at hand from the two fits, can be combined to form test statistics for breakpoints.

Such methods suffer a loss of power because of two defects. First, χ^2's by construction throw away information about the arithmetic signs of differences; such information is relevant since a track's momentum should normally *decrease* when the particle decays.

Second, appropriate constraints of a breakpoint hypothesis are not incorporated. For example, if a particle decays at hit k, a desired quantity is the mismatch in track parameters describing the *momentum* vector, *under the constraint* that the track *position* vector from fits (a) and (b) is identical.

In this paper, we describe a procedure which uses information from the Kalman filter fit to construct the result of a *full* track fit which has additional track parameters to account for the discontinuities at a breakpoint. It is natural to allow for one, two, or three discontinuous parameters in order to describe different physical processes:

- **Type I:** An electron emitting a bremsstrahlung photon generally changes only its momentum *magnitude*, since the photon is essentially collinear with the electron direction.
- **Type II:** A particle with a hard elastic scatter may have momentum magnitude essentially unchanged, while changing the two angles specifying direction.
- **Type III:** A charged pion or kaon decaying to $\mu\nu$ in general changes momentum magnitude as well as the two angles.

With the method we describe, trial breakpoint fits at every hit (away from the ends) of the track can be quickly obtained, and used to search for and characterize breakpoints.

Billoir [1] has investigated Type II breakpoints, performing fits which do not assume an existing breakpointless fit. We show in this paper how the by-products of a Kalman filter fit allow one to avoid refitting the hits while incorporating the constraints.

We discuss these tools in the context of the NOMAD [7] neutrino experiment at CERN, within which this development took place. However, the results are generally applicable to any experiment in which the number of position measurements is large enough to fit the track on both sides of the potential breakpoint.

In Sec. 2 we introduce some notation and describe our track parameters and track models, including energy loss. In Sec. 3 a review of the traditional (non-Kalman) track fit is given. Sec. 4 describes its replacement by the Kalman filter. In Sec. 5, we briefly review previous work on breakpoint variables. In Sec. 6, we introduce the new breakpoint variables, and in Sec. 7, we present some indicative results of their use. We conclude in Sec. 8.

2 Track parameters and track models

2.1 Parameters

When fitting a track, one typically describes its location in 6D phase space by choosing a fixed reference surface (a vacuum window, chamber plane, etc.) and then fitting for the 5 independent parameters of the track at the position *where the track intersects this surface.* We let the vector $\mathbf{x}$ contain these track parameters. In a fixed-target experiment with beam direction along the z axis, the parameterization of $\mathbf{x}$ is often taken to be

$$\mathbf{x} = (x, y, \, \mathrm{d}x/\,\mathrm{d}z, \, \mathrm{d}y/\,\mathrm{d}z, q/p) \tag{1}$$

at a reference plane at fixed z, where q/p is the charge/momentum.

NOMAD is a fixed-target experiment with drift chamber planes perpendicular to the z direction (nearly aligned with the neutrino beam). However, the chambers are immersed in a uniform magnetic field [3], so that soft tracks often loop back, and a helical parameterization similar to collider experiments is

[3] Sense wires of one chamber make angles of +5, 0 and −5 degrees with respect to the magnetic field direction providing a space measurement along coordinates designated u, y, and v.

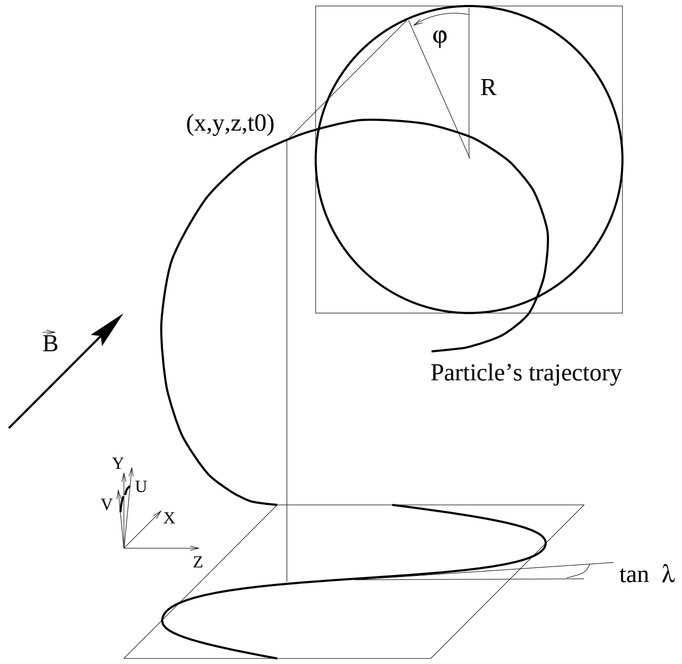

Fig. 1. Definition of the helix parameters used to describe charged particle trajectory in the NOMAD setup.

more appropriate. We maintain the reference surface as a plane with fixed z (the z of the first hit of the track), and specify a track there by

$$\mathbf{x} = (x, y, 1/R, \tan\lambda, \phi, t), \tag{2}$$

where we have introduced the three parameters of a helical curve in the uniform magnetic field: the signed [4] inverse radius of curvature $1/R$, the dip angle $\tan\lambda$, and the angle of rotation ϕ (see Fig. 1). In addition to these 5 traditional parameters, NOMAD has a sixth parameter, the zero-time-offset for the drift chamber measurements, called t. It has been introduced because of trigger time jitters. In the following, unless specified otherwise, *track parameters* will refer to this second parameterization.

The parameter $1/R$ is related to the momentum p by

$$\frac{1}{R} \propto \frac{B}{p}\sqrt{1 + (\tan\lambda)^2}, \tag{3}$$

while the sign of $(1/R)$ only reflects the particle charge if the time runs the right way along the track. In NOMAD the sign convention we used implied that the product $R \cdot \phi$ increases with time along the assumed time direction given by the ordering of the measurements along the track.

[4] In NOMAD, $1/R$ has a sign opposite to the particle charge.

Throughout our work, we use a change in $1/R$ as an indicator of a change in p; this is strictly true only when the change in $\tan\lambda$ is negligible, but is an adequate approximation. (Inhomogeneity in the magnetic field B does not matter, since we compare $1/R$ estimates at the same point.)

With this parameterization, the three physical processes enumerated in the Introduction have the following breakpoint signatures:

Type I: An electron emitting a bremsstrahlung photon has a discontinuity in $1/R$.
Type II: A pion with a hard elastic scatter has discontinuities in $\tan\lambda$ and ϕ.
Type III: A charged pion or kaon decaying to $\mu\nu$ has discontinuities in $1/R$, $\tan\lambda$, and ϕ.

From the estimation of the track parameters at the reference plane one needs a transformation, called the track model, with which one computes the expected measurements at any position in the detector. This model describes the dependence of the measurements on the initial values in the ideal case of no measurement errors and of deterministic interactions of the particle with matter. Use of a correct description is of the utmost importance for the performance of the fitting procedure, be it traditional or not.

2.2 Equation of motion in a magnetic field

The trajectory of a charged particle in a (static) magnetic field is determined by the following equation of motion:

$$\mathrm{d}^2\mathbf{r}/\,\mathrm{d}^2 s = (kq/p)\cdot(\,\mathrm{d}\mathbf{r}/\,\mathrm{d}s)\times\mathbf{B}(\mathbf{r}(s)), \tag{4}$$

where $\mathbf{r}$ is the position vector, s is the path length, k is a constant of proportionality, q is the (signed) charge of the particle, p is the absolute value of its momentum, and $\mathbf{B}(\mathbf{r})$ is the static magnetic field.

With our parameterization it proved convenient to use ϕ as the running parameter rather than z since particles may loop back in the detector and cross the same measurement plane several times. The equation of motion can be readily integrated along a trajectory step (from position 0 to position 1) where R and $\tan\lambda$ are assumed constant.

$$x_1 = x_0 + R_0\cdot\tan\lambda\cdot(\phi_1-\phi_0) \tag{5}$$
$$y_1 = y_0 + R_0\cdot(\cos\phi_1-\cos\phi_0) \tag{6}$$
$$z_1 = z_0 + R_0\cdot(\sin\phi_1-\sin\phi_0) \tag{7}$$

$$
\begin{aligned}
R_1 &= R_0 && (8)\\
\tan\lambda_1 &= \tan\lambda_0 && (9)\\
t_1 &= t_0 + R_0\cdot(\phi_1-\phi_0)/(\beta\cos\lambda_0). && (10)
\end{aligned}
$$

In the last equation, β is the particle velocity and $R\cdot\Delta\phi$ has the right sign following our convention.

In NOMAD, detector measurement planes are located at fixed z and Eqn. 7 can be solved,

$$\sin\phi_1 = \sin\phi_0 + (z_1 - z_0)\cdot(1/R_0), \tag{11}$$

to obtain ϕ_1 at the desired z. In practice, among all the possible solutions, our track model returns the one corresponding to the next crossing in the requested time direction.

The magnetic field strength varies by a few percent in the tracking volume. This was accommodated by ignoring the minor components of the field and updating $1/R$ at every tracking step so that the product $R{\cdot}B$ remains constant, up to energy losses which are now discussed.

2.3 Energy losses

The ionization losses are accounted for by updating $1/R$ at every tracking step [5]:

$$\Delta(1/R) = \frac{\mathrm{d}(1/R)}{\mathrm{d}E}\frac{\mathrm{d}E}{\mathrm{d}x}\Delta x = \frac{|1/R|}{0.3B\beta}\frac{\mathrm{d}E}{\mathrm{d}x}\Delta\phi \tag{12}$$

where $\Delta\phi = \phi_{end} - \phi_{start}$, and where $\mathrm{d}E/\mathrm{d}x$ is given by the Bethe-Bloch equation, evaluated (by default) with the pion mass, and a local matter density extracted from the detector model used in the GEANT [8] simulation of the experiment. In the central tracker part of the detector, the matter density is about 0.1 g/cm^3 : the ionization loss model does not need to include detailed relativistic corrections.

In NOMAD, bremsstrahlung losses should be accounted for (on average) in the track model for electron tracks, because the central tracker amounts to about 1 radiation length (X_0). In the electron (and positron) track model, one adds

[5] The absolute value derives from our sign conventions, so that a track gains energy if tracked backwards in time.

to the ionization losses (evaluated with the electron mass) bremsstrahlung losses:

$$\Delta(1/R) = \frac{\Delta\phi}{\beta^2 X_0 \cos\lambda}, \tag{13}$$

where we can readily approximate $\beta = 1$. As expressed, these losses include the whole radiated photon spectrum, although beyond a certain threshold, the radiated photons can be detected in the downstream electromagnetic calorimeter, or sometimes as a conversion pair in the tracker. But no convincing way was found to define a threshold that separates continuous small losses from accidental big ones.

3 Traditional track fits

Given the track model and an estimate ($\mathbf{x}_0$) of the parameters at the reference position, one can compute at each measurement location in the detector (labeled $k = 1, \ldots N$) the theoretical "*ideal measurements*" which would be made in the absence of fluctuations due to the two categories of "noise": process noise (multiple scattering, bremsstrahlung, etc.) and measurement noise (detector resolution). This computation is represented by the system equation in the absence of noise,

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_0), \tag{14}$$

where $\mathbf{f}$ is a deterministic function (the track model) giving the values of the parameters at each location k.

In general the set of parameters $\mathbf{x}_k$ is not measured directly by the apparatus; only a function of it, $h_k(\mathbf{x}_k)$, is observed. (In our case h_k is a drift chamber position measurement.) Let

$$m_k \;=\; h_k(\mathbf{x}_k) \;+\; \varepsilon_k \tag{15}$$

be the measurement equation where ε_k represents the measurement errors. By convention, $\langle m_k \rangle \;=\; h_k(\mathbf{x}_k)$ and $\langle \varepsilon_i \rangle = 0$. In practice, one has to check that the track model (involved in the calculation of $\mathbf{x}_k$) and the measurement function h_k fulfill this convention.

In a traditional (non-Kalman) track fit, one calculates the $N \times N$ covariance matrix of the measurements [6]:

$$V_{ij}^{(m)} = \langle (m_i - h_i(\mathbf{x}_i))(m_j - h_j(\mathbf{x}_j)) \rangle, \tag{16}$$

where the angle-brackets represent an average over an ensemble of tracks with the same track parameters $\mathbf{x}$. According to the Gauss-Markov theorem, the minimization of

$$\chi^2_{\mathrm{trk}}(\mathbf{x}) = (\mathbf{m} - \mathbf{h}(\mathbf{x}))^T \ [V^{(m)}]^{-1} \ (\mathbf{m} - \mathbf{h}(\mathbf{x})) \tag{17}$$

yields parameter estimates with minimum variance among all linear unbiased estimates.

Note that when the parameter evolution is not affected by any stochastic noise (as assumed by Eq. 14), $V_{ij}^{(m)} = \langle \varepsilon_i \ \varepsilon_j \rangle$ is diagonal (at least block diagonal, if the apparatus provides multidimensional correlated measurements), and may depend weakly on $\mathbf{x}$ via the detector response (for example, the spatial resolution of drift chambers depends on the track angle w.r.t the anode plane).

The system equation becomes non deterministic when the track experiences stochastic processes such as multiple scattering, bremsstrahlung or ionization losses. The system equation (Eq. 14) becomes :

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_0) + \mathbf{w}_k \tag{18}$$

where $\mathbf{w}_k$ is a random vector representing the fluctuation of the parameters along the path from location 0 to location k. This *process noise* translates, via the track model, into off-diagonal elements in $V^{(m)}$ (the noise from location 0 to k depends on the noise from location 0 to $k-1$ and $k-1$ to k).

More importantly, $V^{(m)}$ now depends on the reference location z_0 at which the parameters will be estimated. Of course the average value of the perturbing effects have to be included in $\mathbf{f}_k$ but the fluctuations (e.g. track scattering or energy loss straggling when relevant) have to be described by the p.d.f of $\mathbf{w}_k$. In fact only the covariance matrix of the $\mathbf{w}_k$ is needed in practice. (See Sec. 4.)

Given particular data $\mathbf{m}$, the track fit consists in finding the value of $\mathbf{x}$ which minimizes Eq. 17. Minimization is typically an iterative process with some convergence criteria to decide when to stop iterating. The matrix $V^{(m)}$ can be calculated by Monte Carlo techniques or sometimes analytically. If one is

[6] The superscript in parentheses in $V^{(m)}$ is to make clear which vector it corresponds to; this will be the convention in this paper.

fortunate, the calculation depends only weakly on $\mathbf{x}$, so it can be done once per fit using the initial guess for $\mathbf{x}$, and not changed at each iteration. Note that the $N \times N$ matrix $V^{(m)}$ must be inverted, where N is the number of measurement positions.

Let χ^2_{trk} (written without explicit argument $\mathbf{x}$) be the minimum value after convergence of the minimization procedure and $\widehat{\mathbf{x}}$ the value of the estimated parameters (the value of $\mathbf{x}$ at the minimum). The covariance matrix of the estimates is then approximated by the inverse of the curvature matrix at the minimum :

$$V_{ij}^{(\widehat{\mathbf{x}})} = \left[\frac{1}{2} \frac{\partial^2 \chi^2_{\mathrm{trk}}}{\partial x_i \partial x_j} \right]^{-1} \tag{19}$$

A traditional fit gives the track parameters $\widehat{\mathbf{x}}$ *only* at the *fixed* reference z_0, say at the beginning of the track. In order to find the best-fit track parameters at the *end* of the track, one has to recompute $V^{(m)}$ using the new reference and perform a completely independent fit. Due to multiple scattering, the results of two fits using different reference z_0's are *not* related by the track model, and cannot be obtained from one another. This effect is already present in a perfectly linear ideal case, where the detector resolution does not depend on $\mathbf{x}$. It just reflects the fact that the weights of measurements to estimate the track parameters depend (eventually strongly) on z_0.

This is unfortunate, since in practice, track extrapolation is often desired from both ends of the track. Furthermore, optimal track estimates at every possible sensor position can be quite useful, either to detect outliers efficiently, or to optimally collect hits left over during a first pass.

Finally, if one attempts such a traditional track fit in an experiment with large N (such as NOMAD, where N ranges up to 150) N can be too big to make inversion of $V^{(m)}$ practical.

Given these difficulties, in NOMAD the Kalman filter was implemented instead [9].

4 The Kalman Filter

The Kalman filter is a least-squares stepwise parameter estimation technique. Originally developed in the early 60's to predict rocket trajectories from a set of their past positions, it can be used to handle multiple scattering while estimating track parameters. We try here to briefly shed light on the features of the Kalman filter for track fitting and refer to the literature for more details [1–4].

The Kalman filter technique gives, mathematically speaking, exactly the same result as a standard least squares minimization. In the framework of track fitting, it essentially avoids big matrix inversion and provides almost for free an optimal estimate of track parameters at any location, allowing the detection of outlying measurements, extrapolation and interpolation into other subdetectors.

The set of parameters $\mathbf{x}$ is called the state vector in Kalman filtering.

Starting from Eq. 18 we rewrite the system equation in a stepwise form, where the state vector at location k is obtained from its value at previous location [7] $k-1$.

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{w}_k \tag{20}$$

We shall assume in the following that both $\mathbf{w}_k$ and ε_k (the measurement errors from Eq. 15) are independent random variables with mean 0 and a finite covariance matrix.

Linearizing the system in the vicinity of $\mathbf{x}_{k-1}$, one obtains:

$$\mathbf{f}_k(\mathbf{x}_{k-1}) = F_k \cdot \mathbf{x}_{k-1} \tag{21}$$

$$h_k(\mathbf{x}_k) = H_k \cdot \mathbf{x}_k \tag{22}$$

for the track model and the measurement equation.

We can now recall keywords used in the Kalman filter estimation technique:

- **Prediction** is the estimation of the state vector at a "future" time, that is the estimation of the state vector at time or position $(k+1)$ using all the measurements up to and including m_k.
- **Filtering** is estimating the "present" state vector based upon all present and "past" measurements. For *Forward* filtering, this means estimating track parameters at k using measurements up to and including m_k. For *Backward* filtering, this means estimating track parameters at k using the measurements m_N down to m_k.

[7] One has to assume that the measurements are ordered with respect to time to handle multiple scattering because the covariance matrix of measurement residuals depends on this order. Without multiple scattering, the ordering does not affect the filter result as in the case of a traditional fit.

- **Filter.** The algorithm which performs filtering is called a filter and is built incrementally: filtering m_1 to m_k consists in filtering m_1 to m_{k-1}, propagating the track from m_{k-1} to m_k and including m_k. A filter can proceed forward (k increases) or backward (k decreases).
- **Smoothing** means using all the measurements to provide a track parameter estimate at any position. The smoothed estimate can be obtained as a weighted mean of two filtered estimates: the first one using m_1 to m_k (forward), the other using m_N to m_{k+1} (backward).[8]

One can understand the basic idea of the Kalman filter in the following way. If there is an estimate of the state vector at time (location) t_{k-1}, it is extrapolated to time t_k by means of the system equation. The estimate at time t_k is then computed as the weighted mean of the predicted state vector and of the actual measurement at time t_k, according to the measurement equation. The information contained in this estimate can be passed back to all previous estimates by means of a second filter running backwards or by the smoother.

The main formulas for our linear dynamic system are the following:

System equation:

$$\mathbf{x}_k = F_k \cdot \mathbf{x}_{k-1} + \mathbf{w}_k \tag{23}$$

$$E\{\mathbf{w}_k\} = 0, \quad \mathrm{cov}\{\mathbf{w}_k\} = Q_k \quad (1 \leq k \leq N) \tag{24}$$

Measurement equation:

$$m_k = H_k \cdot \mathbf{x}_k + \varepsilon_k \tag{25}$$

$$E\{\varepsilon_k\} = 0, \quad \mathrm{cov}\{\varepsilon_k\} = V_k = G_k^{-1} \quad (1 \leq k \leq N) \tag{26}$$

where the matrices Q_k and V_k represent the process noise (multiple scattering, bremsstrahlung, etc.) and measurement noise (detector resolution) respectively. The details of Q_k calculation for the parameterization adopted in NOMAD can be found in Ref. [10].

As an example we include here the formulas for making a prediction:

[8] This leads to a subtlety in practice, when we actually have in hand the forward and backward filter estimates at k; averaging these would lead to double-counting the information from m_k. Hence, to be proper, one must unfilter m_k from one of the estimates. This small correction is implemented in our smoother and in the quantity $\chi_k^{2\,(FB)}$ discussed below, but was deemed negligible and never implemented in the other breakpoint quantities.

- Extrapolation of the state vector:
  $\mathbf{x}_k^{k-1} = F_k \mathbf{x}_{k-1}$
- Extrapolation of the covariance matrix:
  $C_k^{k-1} = F_k C_{k-1} F_k^T + Q_k$
- Predicted residuals:
  $\mathbf{r}_k^{k-1} = m_k - H_k \mathbf{x}_k^{k-1}$
- Covariance matrix of the predicted residuals:
  $R_k^{k-1} = V_k + H_k C_k^{k-1} H_k^T$

Using the Kalman filter, the computer time consumed for a track fit is proportional to the number of hits on the track, while with the traditional technique it is proportional to the cube of the same number in case of multiple scattering.

After the Kalman fitting procedure one has the following available information :

- $\widehat{\mathbf{x}}_k^F$ and $\widehat{\mathbf{x}}_k^B$: the Forward and Backward estimates of the state vector at position k, i.e., the estimate of the track parameters at location k using measurements 1 up to k (forward) or N down to k (backward).
- $\chi_k^{2\ (F)}$ and $\chi_k^{2\ (B)}$: the minimum χ^2 value of the forward and backward fits up to measurement k.
- $V^{(\widehat{\mathbf{x}}_k,F)}$ and $V^{(\widehat{\mathbf{x}}_k,B)}$: the covariance matrices of $\widehat{\mathbf{x}}_k^F$ and $\widehat{\mathbf{x}}_k^B$ respectively.
- $\widehat{\mathbf{x}}_k$, $\chi^2_{\mathbf{trk}}$ and $V^{(\widehat{\mathbf{x}}_k)}$: the same quantities, determined from the smoothed estimates, the equivalent of a full fit done at location k. Note that the $\chi^2_{\mathbf{trk}}$ minimum does not depends on the location at which the parameter are estimated ($\chi^2_{\mathbf{trk}}$ for $\widehat{\mathbf{x}}_k$ is independent of k).

Thus, much information exists as the by-product of a track fit: at every hit on the track away from the ends, we have the results of three fits for the track parameters at that hit: a fit to the part of the track upstream, a fit to the part of the track downstream, and a fit to the whole track. This information is the input to breakpoint analyses.

5 Earlier Applications to Breakpoint Searches

A natural way to compare $\widehat{\mathbf{x}}_k^F$ and $\widehat{\mathbf{x}}_k^B$ is discussed by R. Frühwirth [3] and was implemented in NOMAD tracking [9,11] before developing our extensions. One simply constructs the χ^2 of the mismatch of all the forward-backward parameters at each hit k:

$$\chi_k^{2\ (FB)} = (\widehat{\mathbf{x}}_k^B - \widehat{\mathbf{x}}_k^F)^T \ [V^{(\widehat{\mathbf{x}}_k,B)} + V^{(\widehat{\mathbf{x}}_k,F)}]^{-1} \ (\widehat{\mathbf{x}}_k^B - \widehat{\mathbf{x}}_k^F). \tag{27}$$

The value of $\chi_k^{2\ (FB)}$ is easily computed from the following relationship which holds for any k:

$$\chi^2_{\mathrm{trk}} = \chi_k^{2\ (F)} + \chi_k^{2\ (B)} + \chi_k^{2\ (FB)}. \tag{28}$$

After the track fit, we find the hit k for which $\chi_k^{2\ (FB)}$ is a maximum, i.e., for which the forward-backward mismatch in track parameters has greatest significance; we call this maximum value $\tilde{\chi}^{2\ (FB)}$. One can assign a breakpoint at that k there if $\tilde{\chi}^{2\ (FB)}$ is above some threshold. Frühwirth also investigated various combinations of $\tilde{\chi}^{2\ (FB)}$ with $\chi_k^{2\ (F)}$, $\chi_k^{2\ (B)}$, and degrees of freedom in the track fits, but concluded that $\tilde{\chi}^{2\ (FB)}$ was his best breakpoint test statistic [3,12].

6 Some Additional Breakpoint Variables Based on Constrained Fits to the Specific Breakpoint Types

One may suspect that previously defined breakpoint tests do not make optimal use of the available information. Any χ^2 quantity is by definition insensitive to the arithmetic sign of differences, while in the processes of interest, a *decrease* in the momentum is expected. Furthermore, $\chi_k^{2\ (FB)}$ mixes all the parameter mismatch information together. The signed forward-backward mismatch in single quantities such as $1/R$ can be examined, but it has the problem that the other 5 parameters are not constrained to be the same. (Physical changes in $1/R$ can result, for example, in fitted mismatches in ϕ as well as $1/R$.) Finally, an optimal test should use a more fully developed breakpoint hypothesis, so that a more meaningful comparison of χ^2's, with and without breakpoints, can take place.

6.1 Constrained Fits to Breakpoints

Here we show how to obtain and examine the result that one would get by doing a traditional fit which uses *all* hits, *but which allows a subset of the track parameters to be discontinuous* at a particular hit k. We parameterize the full track with 1 to 3 added parameters in order to incorporate breakpoints of Types I, II, and III at hit k. E.g., for Type I, we replace the parameter $1/R$ by two parameters, a forward value $1/R_F$ just before hit k and a back value $1/R_B$ just after hit k. Thus, our fits to the 3 breakpoint types have 7, 8, and 9 parameters, respectively. We denote these sets with breakpoints by $\boldsymbol{\alpha}$ rather than $\mathbf{x}$, and they are, respectively for the three types:

$$\boldsymbol{\alpha}_{\mathrm{I}} = \{x, y, 1/R_F, 1/R_B, \tan\lambda, \phi, t\}, \tag{29}$$
$$\boldsymbol{\alpha}_{\mathrm{II}} = \{x, y, 1/R, \tan\lambda_F, \tan\lambda_B, \phi_F, \phi_B, t\}, \tag{30}$$
$$\boldsymbol{\alpha}_{\mathrm{III}} = \{x, y, 1/R_F, 1/R_B, \tan\lambda_F, \tan\lambda_B, \phi_F, \phi_B, t\}. \tag{31}$$

For definiteness, we discuss here the concept in terms of a Type I breakpoint.

One can imagine a cumbersome procedure whereby one puts a Type I breakpoint at a *particular* hit k, and performs a traditional χ^2 track fit (with 7 parameters in our case) to *all* the hits of the track, minimizing the full Type I track's χ^2, which we call

$$\chi^2_{\mathrm{full,I},k}(\boldsymbol{\alpha}_{\mathrm{I}}). \tag{32}$$

One would obtain the best estimate of $\boldsymbol{\alpha}_{\mathrm{I}}$, its covariance matrix, and the minimum value of $\chi^2_{\mathrm{full,I},k}$, all for a breakpoint at that particular hit k. One could then repeat this for each possible value of k, obtaining numerous potential track-with-breakpoint fits.

Essentially *the same set of results* can be obtained far more economically by starting from the results $\hat{\mathbf{x}}_k^F$ and $\hat{\mathbf{x}}_k^B$ at each k which *already exist* as by-products of the Kalman filter fit. These results carry all the information that we need, since their error matrices contain the information (up to linear approximation) on how $\chi_k^{2\ (F)}$ and $\chi_k^{2\ (B)}$ change when the track parameters change. We need only perform a linear χ^2 minimization in which $\{\hat{\mathbf{x}}_k^F, \hat{\mathbf{x}}_k^B\}$ (now playing the role of the "measured data") is compared to the Type I breakpoint model prediction $H_I\boldsymbol{\alpha}_{\mathrm{I}}$. H_I is the model matrix with 12 rows and 7 columns containing only zeros and ones :

$$H_I\boldsymbol{\alpha}_{\mathrm{I}} = \{(x, y, 1/R_F, \tan\lambda, \phi, t), (x, y, 1/R_B, \tan\lambda, \phi, t)\}. \tag{33}$$

Or, introducing the two 6×7 submatrices H_I^F and H_I^B of H_I :

$$H_I\boldsymbol{\alpha}_{\mathrm{I}} = (H_I^F\boldsymbol{\alpha}_{\mathrm{I}}, H_I^B\boldsymbol{\alpha}_{\mathrm{I}}). \tag{34}$$

Since our 12 pieces of "measured data" $\{\hat{\mathbf{x}}_k^F, \hat{\mathbf{x}}_k^B\}$ are two independent sets of 6 parameters with their corresponding covariance matrices, the appropriate chisquare can be written as :

$$\begin{aligned}\chi_k^{2\ (FB)}(\boldsymbol{\alpha}) &= (\hat{\mathbf{x}}_k^F - H^F\boldsymbol{\alpha})^T\ [V^{(\hat{\mathbf{x}}_k,F)}]^{-1}\ (\hat{\mathbf{x}}_k^F - H^F\boldsymbol{\alpha}) \\ &+ (\hat{\mathbf{x}}_k^B - H^B\boldsymbol{\alpha})^T\ [V^{(\hat{\mathbf{x}}_k,B)}]^{-1}\ (\hat{\mathbf{x}}_k^B - H^B\boldsymbol{\alpha}).\end{aligned} \tag{35}$$

(Here and below, we suppress the subscript I since the equations are true for all breakpoint types, with Eqns. 33 and 34 suitably changed.)

The full χ^2 of Eq. 32 can be written as :

$$\chi^2_{\text{full,I},k}(\boldsymbol{\alpha}) = (\mathbf{m} - \mathbf{h}(H\boldsymbol{\alpha}))^T \left[V_k^{(m)}\right]^{-1} (\mathbf{m} - \mathbf{h}(H\boldsymbol{\alpha})) \tag{36}$$

where $V_k^{(m)}$ is the block diagonal matrix containing the covariance matrix $V_k^{(m,F)}$ of the measurements $(m_1 \ldots m_k)$ and the covariance matrix $V_k^{(m,B)}$ of the measurement $(m_{k+1} \ldots m_N)$.

As shown in Appendix A, $\chi_k^{2\ (FB)}(\boldsymbol{\alpha})$ of Eqn. 35 is related to $\chi^2_{\text{full,I},k}(\boldsymbol{\alpha})$ of Eqn. 36 in a revealing way. At each hit k,

$$\chi^2_{\text{full,I},k}(\boldsymbol{\alpha}) = \chi_k^{2\ (F)} + \chi_k^{2\ (B)} + \chi_k^{2\ (FB)}(\boldsymbol{\alpha}), \tag{37}$$

subject to sufficient linearity in the fits. Thus by finding the minimum of $\chi_k^{2\ (FB)}(\boldsymbol{\alpha})$, we find the minimum of $\chi^2_{\text{full,I},k}(\boldsymbol{\alpha})$, since $\chi_k^{2\ (F)}$ and $\chi_k^{2\ (B)}$ are known.

The minimum of $\chi_k^{2\ (FB)}(\boldsymbol{\alpha})$ is obtained *without iteration* since the model relating $\boldsymbol{\alpha}$ to $\{\widehat{\mathbf{x}}_k^F, \widehat{\mathbf{x}}_k^B\}$ is linear. The set of estimated parameters is given by :

$$\widehat{\boldsymbol{\alpha}}_k = V^{(\widehat{\boldsymbol{\alpha}}_k)} H^T (V_k^{(\widehat{\mathbf{x}})})^{-1} \mathbf{X} \tag{38}$$

where $\mathbf{X} = \{\widehat{\mathbf{x}}_k^F, \widehat{\mathbf{x}}_k^B\}$; $V_k^{(\widehat{\mathbf{x}})}$ is the block diagonal matrix containing $V^{(\widehat{\mathbf{x}}_k,F)}$ and $V^{(\widehat{\mathbf{x}}_k,B)}$; and where the covariance matrix for the new estimate, $V^{(\widehat{\boldsymbol{\alpha}}_k)}$, is given by :

$$V^{(\widehat{\boldsymbol{\alpha}}_k)} = \left[H^T (V_k^{(\widehat{\mathbf{x}})})^{-1} H\right]^{-1}. \tag{39}$$

The value of $\chi_k^{2\ (FB)}$ at its minimum is

$$\chi_k^{2\ (FB)}(\widehat{\boldsymbol{\alpha}}) = -\left(H^T (V_k^{(\widehat{\mathbf{x}})})^{-1} \mathbf{X}\right)^T \cdot \widehat{\boldsymbol{\alpha}}. \tag{40}$$

Since the elements of H are mostly 0, and the rest equal to 1, the multiplications by H and H^T were done by hand before coding the software; elements of $(V_k^{(\widehat{\mathbf{x}})})^{-1}$ have only one or two simple terms.

The computer time to perform the Type I, II, and III breakpoint fits at all hits k (away from the track ends) was a negligible addition (few per cent) to the NOMAD track finding and fitting software. This added time was more than paid back by the speedup in matrix inversion which was obtained by explicitly unrolling the loops in the DSINV routine from CERNLIB [13].

From the wealth of information thus available at each hit, we discuss two of the most useful categories: 1) signed differences, in sigma, of breakpoint parameters, and 2) χ^2 comparisons based on the Fisher F-test.

As an example, for Type III breakpoint fits, we let $D_{\mathrm{III},k}(1/R)$ be the forward-backward difference in $1/R$, divided by its standard deviation (sigma, computed from the covariance matrix taking account of errors in both quantities and their correlations). This *signed* quantity effectively gives the significance, in sigma, of the jump in momentum at that hit. Similar quantities, with analogous notation, are calculated for all components of $\boldsymbol{\alpha}$, for all breakpoint Types. Thus, for bremsstrahlung studies, $D_{\mathrm{I},k}(1/R)$ gives the momentum change *under the constraint that all other track parameters are continuous at hit k.*

The *Fisher F statistic* [14] is appropriate for testing if adding parameters yields a statistically significant reduction in the χ^2 of a fit. It is simply the ratio of the respective χ^2/dof for the two versions of the fits. Thus we naturally apply it to our track fits with and without breakpoint parameters. For example, for NOMAD's Type I fits, we have at each hit,

$$F_{\mathrm{I},k} = (\chi^2_{\mathrm{full,I},k}/(N-7)) \Big/ (\chi^2_{\mathrm{trk}}/(N-6)); \qquad (41)$$

we similarly define $F_{\mathrm{II},k}$ and $F_{\mathrm{III},k}$.

For a true breakpoint at a *given* hit k, each F statistic in principle obeys the standard significance-table distributions [14]. However, we normally search through all hits in a track for the lowest value of F, denoted $\widetilde{F}$. This lowest value does not of course follow the usual significance-table values. Hence, as a practical matter, we use data or Monte Carlo events to measure the distribution of $\widetilde{F}$ and the effect of tests based on it.

7 Effectiveness of the additional breakpoint variables

The effectiveness of any breakpoint search is of course highly dependent on details of hardware (e.g., number and quality of the position measurements) and software (e.g., how many tracks with breakpoints are reconstructed as a single track). We present for illustration some experience with the NOMAD detector, using both simulated and real data events.

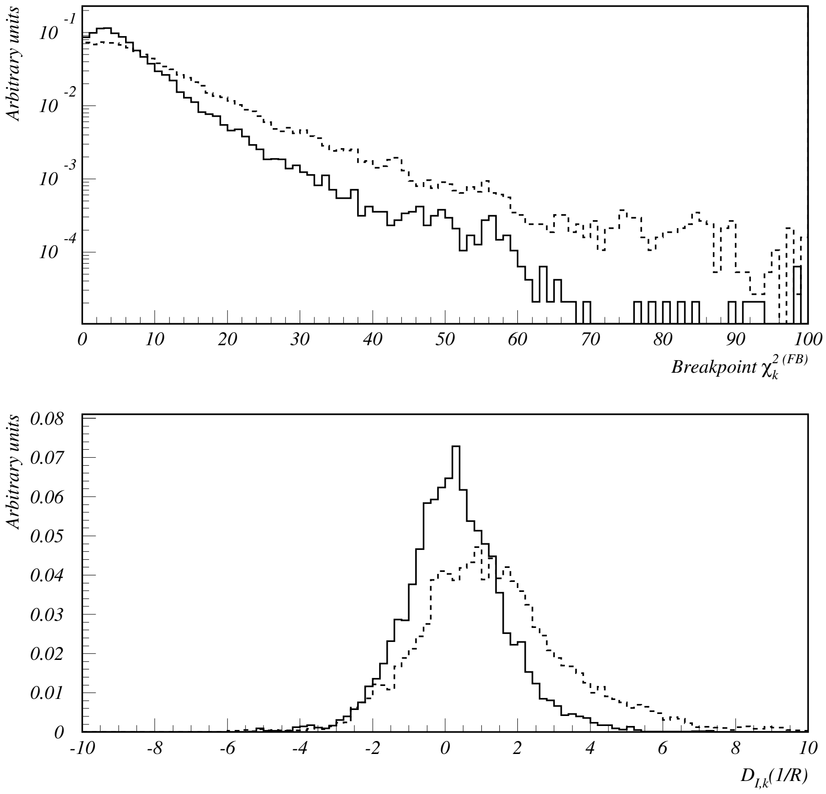

Fig. 2. Test of breakpoint search criteria using real data (muons producing δ-electrons in the NOMAD setup). Comparison of breakpoint chisquare $(\chi_k^{2\ (FB)})$ and normalized difference between curvatures in backward and forward directions $((1/R_B - 1/R_F)/(\sigma_{1/R}))$ for muons (solid line) and electrons (dashed line). In both electron distributions, the excess on the right is evidence of potential breakpoints.

7.1 Electron identification and reconstruction

Algorithms developed for electron identification and reconstruction have been checked under running conditions using δ-electrons produced by straight-through muons (5 GeV $< p_\mu <$ 50 GeV) crossing the NOMAD detector during slow-extracted beam between neutrino spills. This sample of selected electrons from real data can be used to check the subdetector responses compared to simulations and to tune breakpoint search criteria taking into account the effect of the drift chambers alignment quality (see Fig. 2).

A special approach to deal with electrons emitting bremsstrahlung photons (Type I breakpoints) in the NOMAD detector has been developed [10]. If one has identified a reconstructed track in drift chambers as being an electron [9],

[9] Transition Radiation Detector (TRD) is used for electron identification in the

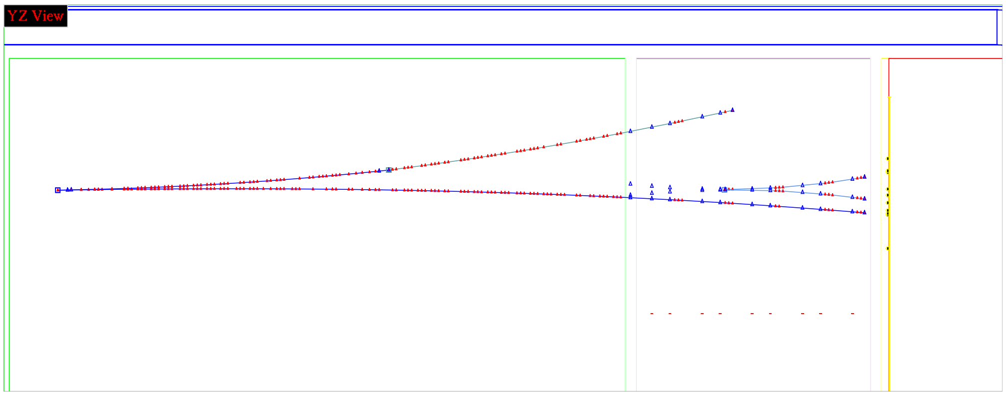

Fig. 3. A reconstructed event from real data (the projection onto the yz plane, in which tracks bend) before an attempt to apply the breakpoint search algorithm. The track at the bottom was identified as an electron by TRD. The triangles are track extrapolations used to search for more hits and to match information from different subdetectors.

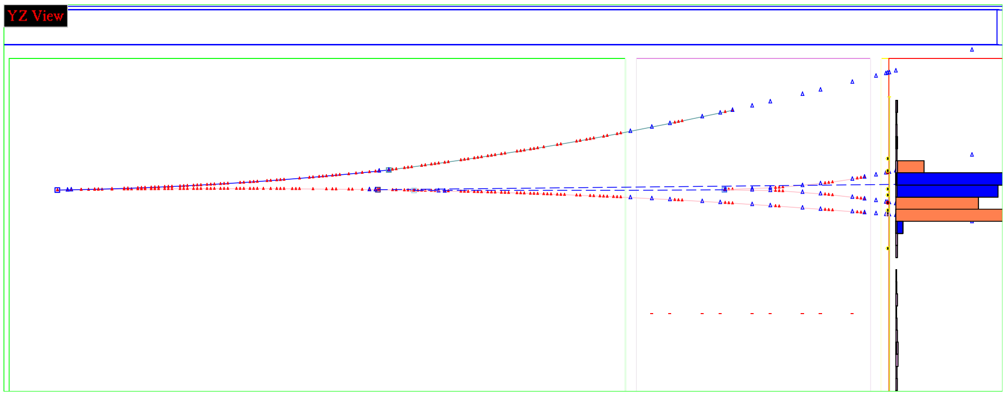

Fig. 4. The same event as before after applying a recursive breakpoint search algorithm. Two breakpoints are found along the electron trajectory and they are associated with two photons (dashed lines): one built out of a conversion inside the drift chambers fiducial volume and the other from a stand-alone cluster in electromagnetic calorimeter. The bars on the right are proportional to energy deposition in the electromagnetic calorimeter.

then hard bremsstrahlung photons can be looked for and neutral tracks can be created requiring further matching with preshower and electromagnetic calorimeter. A successful application of this approach to an event from real data is shown in Fig. 3 and Fig. 4. A recursive breakpoint search algorithm has been applied to a track identified as an electron. As a result two breakpoints

NOMAD experiment.

have been found, each associated with an observed hard bremsstrahlung photon (one of which converts to a reconstructed e^+e^- pair). When applying the breakpoint search algorithm to an electron track one must keep in mind a potential problem related to a possible presence of several kinks along the trajectory (since it can bias the calculation of variables used for the breakpoint search). Details are in Ref. [10].

7.2 Studies using simulated pion decays

As another example of using the breakpoint variables described in Sec. 6.2, we compare several tests for detecting the Type III breakpoint (discontinuity of $1/R$, $\tan\lambda$, and ϕ) of the decay $\pi \to \mu\nu$, and for locating the position of the decay.

The results shown here are for Monte Carlo (MC) simulation [7] of muon neutrino interactions in the NOMAD detector. From the collaboration's standard MC samples, we selected two samples of reconstructed tracks:

- 25000 pions which did not decay and which were reconstructed.
- 2000 pions which did decay via $\pi \to \mu\nu$, and for which a *single* track was reconstructed consisting of hits left by *both* the pion and the muon.

Neither of these two samples contains the pion decays which were broken into separate tracks by the track-finding algorithm, with a vertex assigned at the decay point. The selected tracks were chosen to have more than 20 hits ($N > 20$) and no backward looping, and with the MC pion decays within these hits. Track-finding mistakes (for example use of hits from other tracks in the event) were included, but to obtain the pion track, we required that at least 90% of the hits be correctly assigned.

Figure 5 contains histograms of momentum$\times$charge for the two samples. In order to have samples with similar momenta, we consider here only tracks with momentum less than 6 GeV.

Figure 6 shows, for one of the decaying negative pion tracks, the values of $\chi_k^{2\,(FB)}$, $F_{\mathrm{III},k}$, $D_{\mathrm{III},k}(1/R)$ and $D_{\mathrm{III},k}(\phi)$ at every hit where they are computed. They are plotted at the z-positions of the hits, which in the NOMAD detector are every few cm. The decay point in the MC is indicated by the line at 360 cm. The dotted lines are at the z-positions of the extreme values (maximum or minimum, as relevant for a breakpoint) of the respective variables. Near the MC decay point, both $\chi_k^{2\,(FB)}$ and $F_{\mathrm{III},k}$ reach their extreme values, indicating respectively: a large forward-backward mismatch in the Kalman fits, and a marked improvement in the track χ^2 by adding a three-parameter breakpoint. The sign of the change in $1/R$ corresponds to a decrease in momentum, as

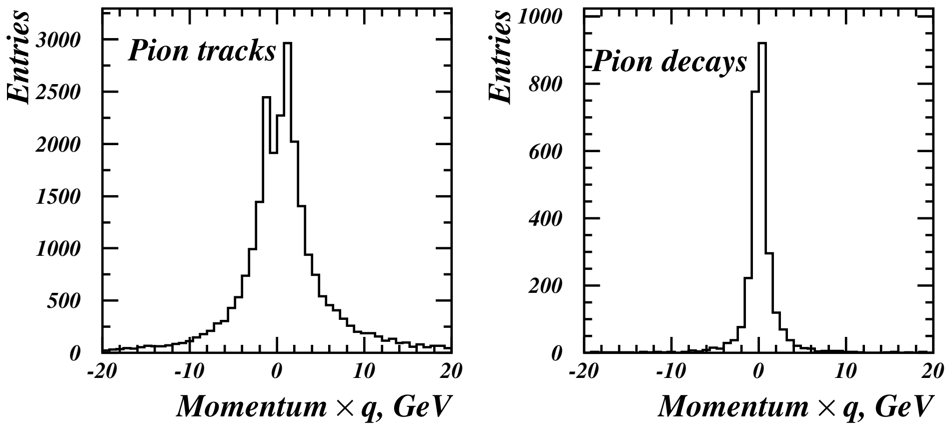

Fig. 5. Histogram of momentum×charge for simulated pions without decay (left) and with decay (right).

expected in a decay. $D_{\mathrm{III},k}(\phi)$ also shows an extremum near the breakpoint; however, we have generally not used it as the primary breakpoint indicator.

7.3 Tests for Existence of a Breakpoint

For testing the existence of a breakpoint, we compare the following test statistics:

- χ^2/dof for the breakpointless fit. (This test gives no additional information on the location of the breakpoint.)
- $\widetilde{\chi}^{2\ (FB)}$. The breakpoint is located at the maximum value of the forward-backward mismatch chisquare ($\chi_k^{2\ (FB)}$) for breakpointless fit among hits in the track;
- $\widetilde{F}$. The breakpoint is located at the minimum value Fisher F among hits in the track (applicable separately to the different breakpoint types if desired).
- $\widetilde{F}_{\mathrm{III}}$ combined with the forward-backward mismatch in radius of the curvature $D_{\mathrm{III},k}(1/R)$ or angle $D_{\mathrm{III},k}(\phi)$. $\widetilde{F}_{\mathrm{III}}$ gives the breakpoint location and $D_{\mathrm{III},k}(1/R)$, $D_{\mathrm{III},k}(\phi)$ are computed at that location.

Fig. 7 contains histograms of χ^2_{trk}/dof, $\widetilde{\chi}^2_{\mathrm{full,I}}$/dof, $\widetilde{\chi}^2_{\mathrm{full,II}}$/dof, and $\widetilde{\chi}^2_{\mathrm{full,III}}$/dof for decaying and non-decaying pions, with the means shown. The χ^2/dof is in general reduced by adding breakpoint parameters. A measure of the significance of the improvement is the Fisher F (Eq. 41), which is shown in Fig. 8 for the same fits. Also shown in Fig. 8 is a histogram of $\widetilde{\chi}^{2\ (FB)}$, the maximum forward-backward mismatch from the breakpointless fit.

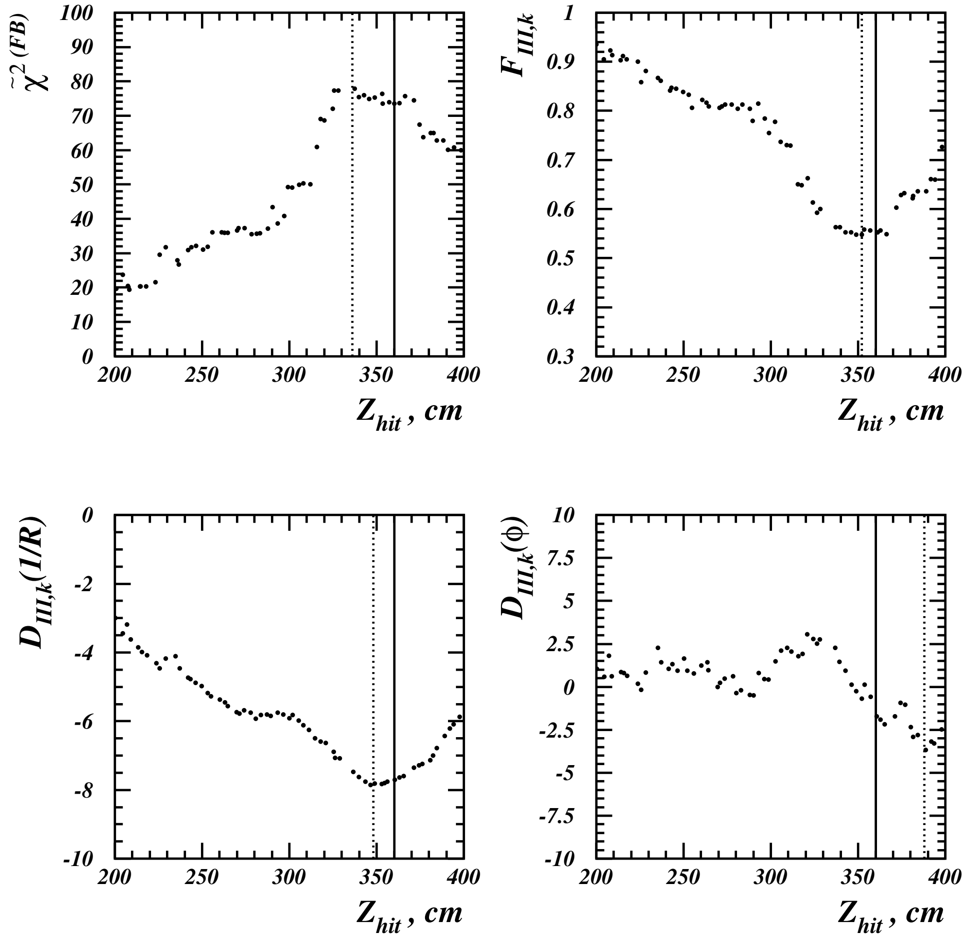

Fig. 6. $\tilde{\chi}^{2\ (FB)}$, $F_{\mathrm{III},k}$, $D_{\mathrm{III},k}(1/R)$ and $D_{\mathrm{III},k}(\phi)$ as a function of the z of the DC hit. The MC decay point is shown in solid line.

From the histograms in Fig. 8, one can calculate the efficiency of labeling tracks as pion decays using "cuts" on these variables. For tabulating a comparison, we choose a cut value for $\widetilde{F}$ such that 10% of non-decaying pions are falsely called pion decays. We then find the efficiency for identifying real pion decays, i.e., the percentage of pion decay tracks which are called pion decays when using the same cut value. This cut value in principle depends on the dof, but for illustration we use the same cut value for all track lengths. The results are given in the first five columns of Table 1. We include for comparison the result for testing for pion decay simply by using $\chi^2_{\mathrm{trk}}/\mathrm{dof}$, i.e., the test one would naturally use if one had only a traditional non-Kalman track fit. As observed by Frühwirth [3], this test is quite competitive with the $\tilde{\chi}^{2\ (FB)}$-test for detecting the existence of a breakpoint, even though it gives no information about the location. The highest efficiency is obtained using $\widetilde{F}_{\mathrm{III}}$. We find this to be true for various other comparisons, although one is cautioned that the

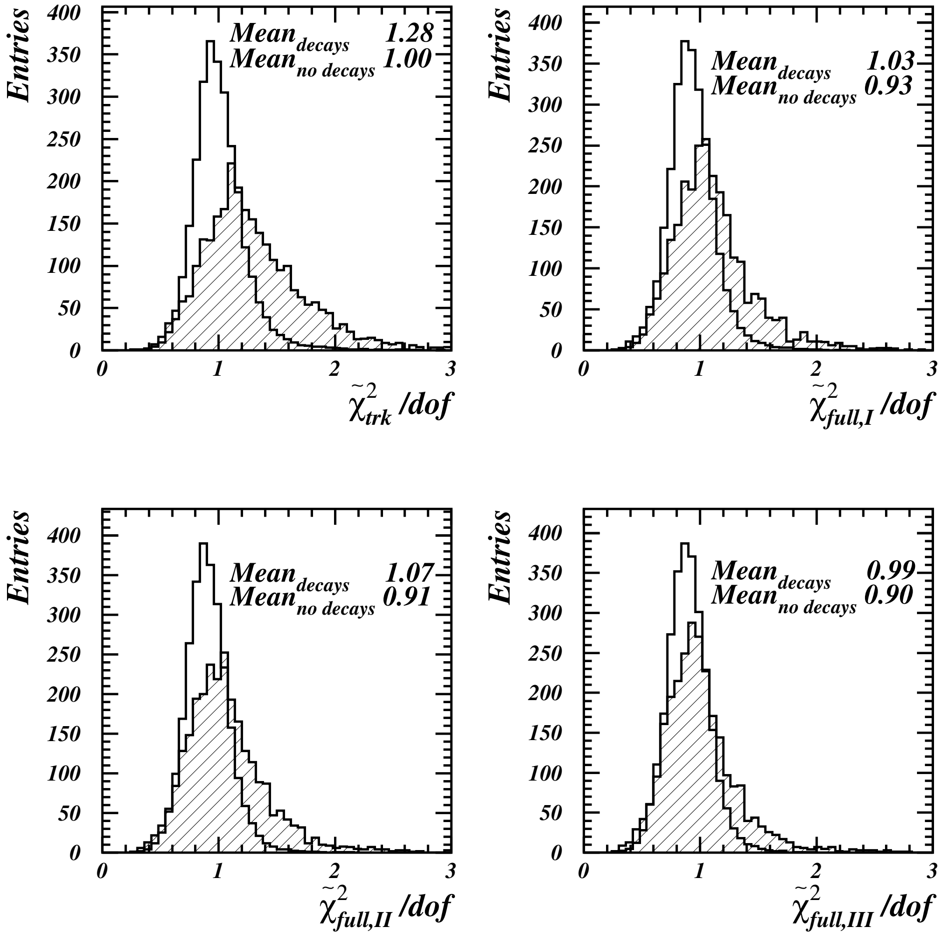

Fig. 7. Histograms of $\chi^2_{\rm trk}/{\rm dof}$, $\tilde{\chi}^2_{\rm full,I}/{\rm dof}$, $\tilde{\chi}^2_{\rm full,II}/{\rm dof}$, and $\tilde{\chi}^2_{\rm full,III}/{\rm dof}$ for the sample of pion decays (shaded) and pions with no decay (white). The samples are normalized to the same number of events.

particular efficiencies listed are for the sample of decaying pions which were not detected by the standard track finding/fitting, and are hence expected to be highly experiment-dependent.

Next we add the signed information available in the new fits: the difference in radius of curvature $D_{{\rm III},k}(1/R)$, and/or the angular differences $D_{{\rm III},k}(\phi)$ and $D_{{\rm III},k}(\tan\lambda)$. We recommend that $D_{{\rm III},k}(1/R)$ not be used to locate the breakpoint, but rather we evaluate this difference at the location dictated by $\widetilde{F}_{\rm III}$. Fig. 9 contains scatter plots of $\widetilde{F}_{\rm III}$ vs $D_{{\rm III},k}(1/R)$ for decays and non-decays. Because true decays have a decrease in momentum, a judicious cut on this scatter plot is more effective than a cut solely on $\widetilde{F}_{\rm III}$. An even more powerful technique is to construct likelihood functions based on these 2D densities, and use the constructed likelihood ratio as a test of pion decay. The

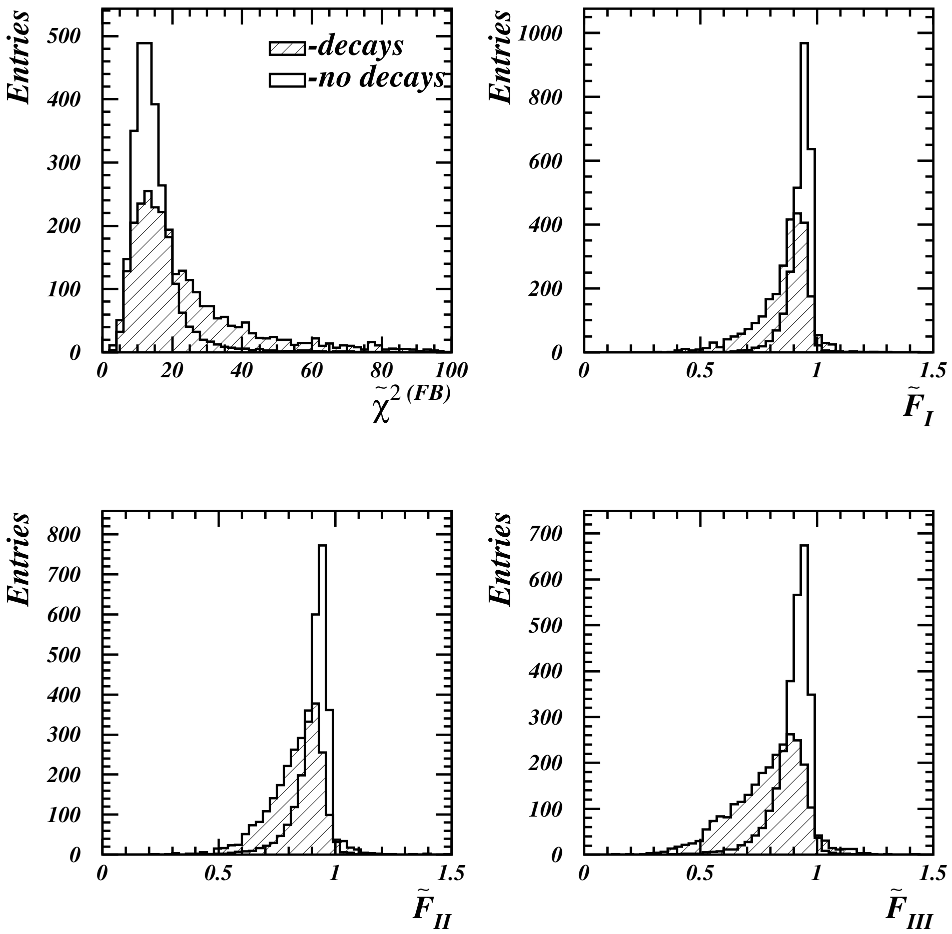

Fig. 8. Histograms of the $\tilde{\chi}^{2\ (FB)}$ and $\tilde{F}_{\mathrm{I}}$, $\tilde{F}_{\mathrm{II}}$, $\tilde{F}_{\mathrm{III}}$ for pion decays (shaded) and pions with no decay (white).

Table 1

Efficiency of the detection of pion decays, using the various test statistics. The first five columns are for simple cuts on the respective variables. The last column is for a cut on a 2D likelihood-ratio test for $\tilde{F}_{\mathrm{III}}$ vs $D_{\mathrm{III},k}(1/R)$ described in the text. The cut value for each test is chosen so that 10% of non-decaying pions are wrongly called decays. The efficiency is computed with respect to the sample which contains only tracks which were not detected as decaying by the original track-finding/fitting algorithms.

χ^2_{trk}/dof	$\tilde{\chi}^{2\ (FB)}$	$\tilde{F}_{\mathrm{I}}$	$\tilde{F}_{\mathrm{II}}$	$\tilde{F}_{\mathrm{III}}$	$\tilde{F}_{\mathrm{III}}$ vs $D_{\mathrm{III},k}(1/R)$
45%	41%	38%	40%	49%	56%

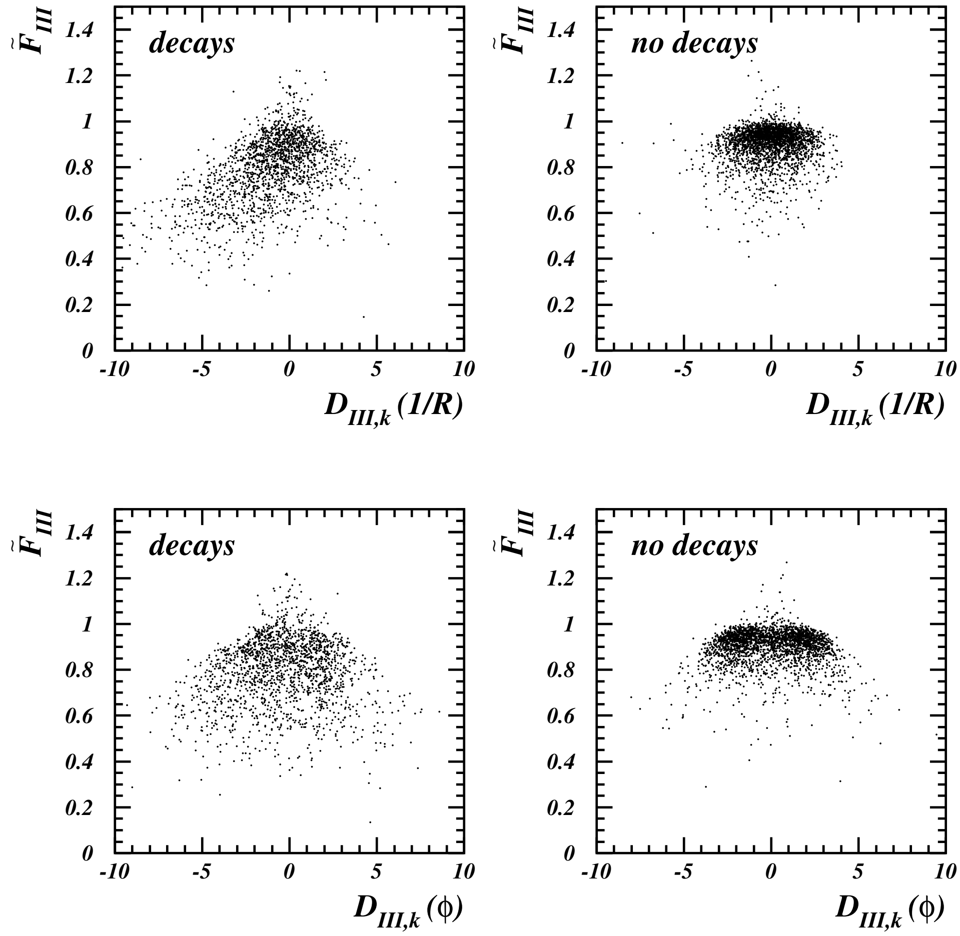

Fig. 9. $\widetilde{F}_{\rm III}$ vs $D_{{\rm III},k}(1/R)$ and $\widetilde{F}_{\rm III}$ vs $D_{{\rm III},k}(\phi)$ histograms for the pion decays and pion with no decays simulated tracks. In order to superimpose negative and positive tracks on the same plots, we have changed the sign of $D_{{\rm III},k}(1/R)$ for one charge. (Recall that 1/R is a signed quantity reflecting the charge of the track.)

result of such a procedure (applied with simple smoothing of the 2D densities, and tested on an independent sample) is given in the last column of Table 1. The improvement over previous breakpoint tests [3] is most significant. Nearly as good efficiencies are obtained from scatter plots of $\widetilde{F}_{\rm III}$ vs $D_{{\rm III},k}(\phi)$ (also shown in Fig. 9), and from $\widetilde{F}_{\rm III}$ vs $D_{{\rm III},k}(\tan\lambda)$.

7.4 Finding the Location of the Breakpoint

One may also ask which of the variables gives the best determination of the location of the breakpoint. We studied in particular the difference between the

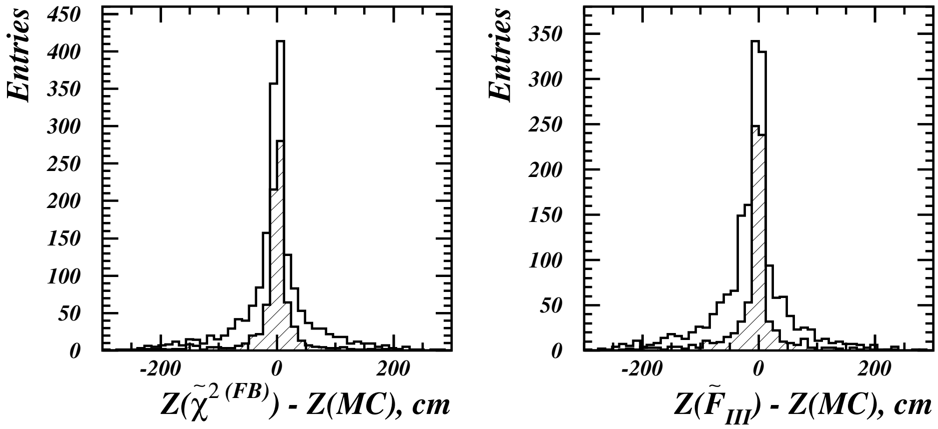

Fig. 10. $\tilde{\chi}^{2\ (FB)}$ and $\tilde{F}_{\rm III}$ resolutions for MC pion decay sample in z. The white histograms are for the initial sample; the shaded histograms are for events in which detected breakpoints passed the selection criteria.

MC decay point and the z position of the hit corresponding to the extrema $\tilde{\chi}^{2\ (FB)}$ or $\tilde{F}_{\rm III}$. These are histogrammed in Fig. 10 for pions which decay. The white histograms show the resolution for the initial samples, with no selection made using $\tilde{\chi}^{2\ (FB)}$ or ($\tilde{F}_{\rm III}$ vs $D_{{\rm III},k}(1/R)$). The shaded histograms show the (more relevant) resolutions for tracks remaining after decay selection using the respective variables. There was not a significant difference, within the limited scope of this study.

8 Conclusions

Replacement of mismatch chisquare for all the forward-backward parameters by the breakpoint variables introduced in Sec. 6.2 can give added power to breakpoint detection in the framework of Kalman filtering technique. We show in particular above that this is the case in a realistic simulation of pion decays in the NOMAD detector. In addition, these breakpoint variables have been successfully used to reconstruct electron hard bremsstrahlung in real data. As expected on theoretical grounds, our most powerful breakpoint detection is based on a scatter plot of a Fisher F-test vs. an appropriate signed difference of a track parameter across the breakpoint.

Acknowledgements

This work was performed within the NOMAD collaboration and hence benefited from numerous aspects of NOMAD's simulation and event reconstruction codes. The authors are grateful to Emmanuel Gangler, Kyan Schahmaneche, and Jean Gosset for their contributions to the NOMAD Kalman filter. We gratefully acknowledge some early investigations by Mai Vo [11] on track breakpoints in NOMAD.

References

[1] P. Billoir, Nucl. Instr. Meth. 225, 352 (1984);

[2] P. Billoir, R. Frühwirth and M. Regler, Nucl. Instr. Meth. A241 (1985) 115-131.

[3] F. Frühwirth, "Application of Filter Methods to the Reconstruction of Tracks and Vertices in Events of Experimental High Energy Physics", HEPHY-PUB 516/88 (Vienna, Dec. 1988)

[4] P. Billoir and S. Qian, Nucl. Instr. Meth. A294 (1990) 219-228; P. Billoir and S. Qian, Nucl. Instrum. Methods A295 (1990) 492-500.

[5] D. Stampfer, M. Regler and R. Frühwirth, "Track fitting with energy loss," Comput. Phys. Commun. 79 (1994) 157-164.

[6] M. Regler, R. Frühwirth and W. Mitaroff, "Filter methods in track and vertex reconstruction," J. Phys. G G22, 521 (1996).

[7] NOMAD Collaboration, J. Altegoer et al., Nucl. Instr. and Meth. A 404 (1998) 96; NOMAD Collaboration, J. Altegoer, et al., Phys. Lett. B 431 (1998) 219; NOMAD Collaboration, P. Astier, et al., Phys. Lett. B 453 (1999) 169.

[8] GEANT : Detector Description and Simulation Tool, CERN Programming Library Long Writeup W5013, GEANT version 3.21

[9] P. Astier, A. Letessier-Selvon, B. Popov, M. Serrano, "NOMAD Reconstruction Software: Drift Chamber Package", version 5 release 2, unpublished (1994), and later releases with additional authors.

[10] Boris Popov, Ph.D. thesis, U. Paris VII, (unpublished, 1998).
http://www-lpnhep.in2p3.fr/Thesards/lestheses.html
http://nuweb.jinr.ru/~popov

[11] Mai Vo, NOMAD collaboration communication (unpublished, 1995) and Ph.D. thesis, Saclay, (unpublished, 1996).

[12] Frühwirth refers to $\tilde{\chi}^{2\,(FB)}$ as C_k. He defines a related quantity, $F_k = (C_k/n_1)/((\chi_k^{2\,(F)} + \chi_k^{2\,(B)})/n_2)$, where n_1 is the d.o.f. for $\tilde{\chi}^{2\,(FB)}$ and n_2 is

the sum of the d.o.f. for $\chi_k^{2\ (F)}$ and $\chi_k^{2\ (B)}$. Frühwirth concludes that the F_k-test is less powerful than the C_k-test, and that the χ^2 is almost as good an indicator of the existence of a kink as the C_k-test.

[13] CERN Program Library routine number F012. A modified version of the routine was used to write the source code of the unrolled loops.

[14] P.R. Bevington and D.K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*, (New York: McGraw-Hill, 1992), pp. 205-209, 261-267.

A Derivation of Eq. 37: $\chi^2_{\mathrm{full,I},k} = \chi_k^{2\ (F)} + \chi_k^{2\ (B)} + \chi_k^{2\ (FB)}$

In Eqn. 36, $V_k^{(m)}$ is the block diagonal matrix containing the covariance matrix $V_k^{(m,F)}$ of the first k measurements $\mathbf{m}^F = (m_1 \dots m_k)$ and the covariance matrix $V_k^{(m,B)}$ of the last $N-k$ measurements $\mathbf{m}^B = (m_{k+1} \dots m_N)$. The right-hand side of Eqn. 36 can thus be split into two terms:

$$\begin{aligned}\chi^2_{\mathrm{full,I},k}(\boldsymbol{\alpha}) = &\ \left[\mathbf{m}^F - \mathbf{h}(H^F\boldsymbol{\alpha})\right]^T\ [V^{(m,F)}]^{-1}\left[\mathbf{m}^F - \mathbf{h}(H^F\boldsymbol{\alpha})\right] \\ &+ \left[\mathbf{m}^B - \mathbf{h}(H^B\boldsymbol{\alpha})\right]^T\ [V^{(m,B)}]^{-1}\left[\mathbf{m}^B - \mathbf{h}(H^B\boldsymbol{\alpha})\right], \end{aligned} \tag{A.1}$$

where one recognizes, in analogy with Eqn. 17, the forward and backward χ^2 terms. We can expand each around their respective minima $\hat{\mathbf{x}}_k^F$ and $\hat{\mathbf{x}}_k^B$, and recall that covariance matrices are the inverse of curvature matrices, giving :

$$\begin{aligned}\chi^2_{\mathrm{full,I},k}(\boldsymbol{\alpha}) = &\ \chi_k^{2\ (F)} + (\Delta\mathbf{x}^F)^T\ [V^{(\hat{\mathbf{x}}_k,F)}]^{-1}\ \Delta\mathbf{x}^F \\ &+\chi_k^{2\ (B)} + (\Delta\mathbf{x}^B)^T\ [V^{(\hat{\mathbf{x}}_k,B)}]^{-1}\ \Delta\mathbf{x}^B, \end{aligned} \tag{A.2}$$

where $\Delta\mathbf{x}^F = \hat{\mathbf{x}}_k^F - H^F\boldsymbol{\alpha}$ and $\Delta\mathbf{x}^B = \hat{\mathbf{x}}_k^B - H^B\boldsymbol{\alpha}$.
Combining this with Eqn. 35 yields Eq. 37.