Nuclear Instruments and Methods in Physics Research A294 (1990) 219–228
North-Holland

SIMULTANEOUS PATTERN RECOGNITION AND TRACK FITTING BY THE KALMAN FILTERING METHOD

P. BILLOIR *

LPC, Collège de France, Paris, France

S. QIAN **

CERN, Geneva, Switzerland
INFN, Frascati, Italy

Received 6 March 1990

A progressive pattern recognition algorithm based on the Kalman filtering method has been tested. The algorithm starts from a small track segment or from a fitted track of a neighbouring detector, then extends the candidate tracks by adding measured points one by one. The fitted parameters and weight matrix of the candidate track are updated when adding a point, and give an increasing precision on prediction of the next point. Thus, pattern recognition and track fitting can be accomplished simultaneously. The method has been implemented and tested for track reconstruction for the vertex detector of the ZEUS experiment at DESY. Detailed procedures of the method and its performance are presented. Its flexibility is described as well.

1. Introduction

The tasks for charged-track reconstruction in experimental high energy physics are pattern recognition (i.e. track finding), track fitting and track matching between detector modules. The Kalman filtering method [1,2] provides a means to do the pattern recognition and track fitting simultaneously [3], and also a natural way to prepare for the track matching. As shown in refs. [4,5], the multiple scattering can be handled properly by the method too.

In brief, the Kalman filtering formalism allows a progressive track-following method. Unlike the traditional pattern recognition, it always carried and updates the information (i.e. track parameters and weight matrix which is the inverse of the error matrix [6]) of candidate tracks along the track-finding process, and gives a precise prediction of the next point to be found. Therefore, when finishing the procedure of pattern recognition, the fitted track parameters are obtained without further computation. At the same time, the fitted tracks are ready to match with tracks from neighbour detectors.

Its speed is faster than of many traditional track-following methods because: (1) combinatorial explosion at high track multiplicities is avoided (the precision of the prediction reduces ambiguities in most cases); (2) we consider the coordinates as linear functions of the track parameters in the neighbourhood of fitted values, so that the minimization of χ^2 leads to the task of solving a system of few linear equations, and needs no iteration. The consumed time is roughly proportional to the track multiplicity; for most events with average track multiplicity of 10 to 20, it is about 10–20 ms CPU time per reconstructed track, on a VAX8800 in our case. In addition, more time and a greater amount of program codes can be saved because the method needs no separate track fitting.

In this article, section 2 describes the implementation of the Kalman filtering method for the vertex detector (VXD) of the ZEUS experiment, step by step. Section 3 includes the procedures of result analysis and shows the performance of the method. Section 4 lists some flexibilities which the method can offer. The conclusions are drawn in the last section.

2. Procedures

The ZEUS VXD has 120 cells around the beam line (fig. 1), each cell has 12 sense wires (fig. 2) parallel to the magnetic field $\boldsymbol{B}$ (of 18 kG) in the z-direction [7]. It does not measure z, so the situation is simplified to a two-dimensional problem.

Projections (in the x–y plane) of all tracks are circles, so the track parameters are chosen as $\boldsymbol{P}_n = (\phi,$

* Now at LPNHE, Universités Paris VI et VII, Paris, France.
** Now in LAA group at CERN, Geneva, Switzerland.

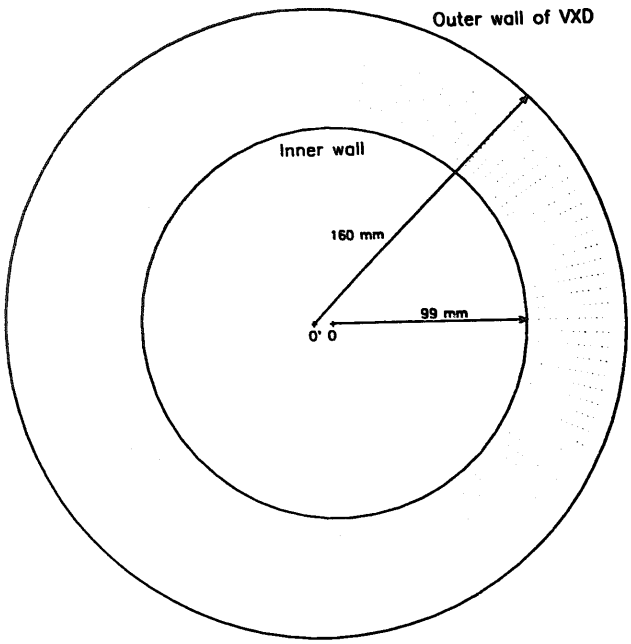

Fig. 1. Global structure of ZEUS VXD in $x-y$ projection. Dotted lines: walls of sensitive cells; O: centre of beam pipe; O': designed beam location which is 10 mm apart from O.

$\sin\beta, 1/R_{tr}$), see fig. 3a, at a fixed radius r_n where the nth sense wire (i.e. the nth measurement plane) is located. ϕ is the azimuthal angle; β is the angle between the ϕ-direction and the tangent of the track at r_n; R_{tr} is the radius of curvature of the track and is proportional to the transverse momentum p_t.

For the case of zero magnetic field B, track parametrization is linear and detailed formulae are listed in appendix A.

Fig. 4 shows the track-following process, where x_n^m denotes the measurement coordinates at the nth wire layer; x_n^F denotes the coordinates corresponding to a fitted track at the nth wire layer; x_n^{Fe} denotes the coordinates corresponding to the extrapolation of x_n^F at the $(n+1)$th wire layer; 1), 2), ··· 6) and 7) label the steps that are described in the following.

2.1. Starting tracks

There may be several ways to start tracks. The one being used now is to first consider all points on the

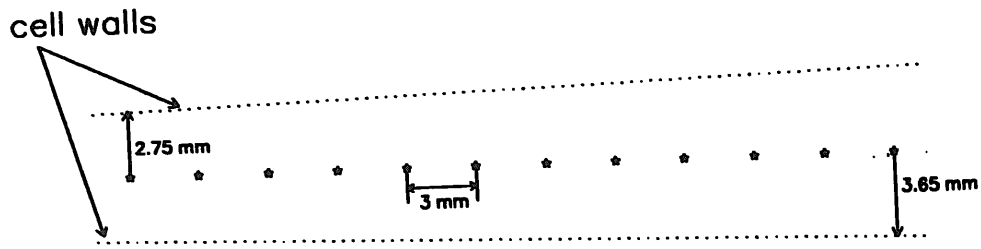

Fig. 2. Dimensions of a sensitive cell. * indicates a sense wire.

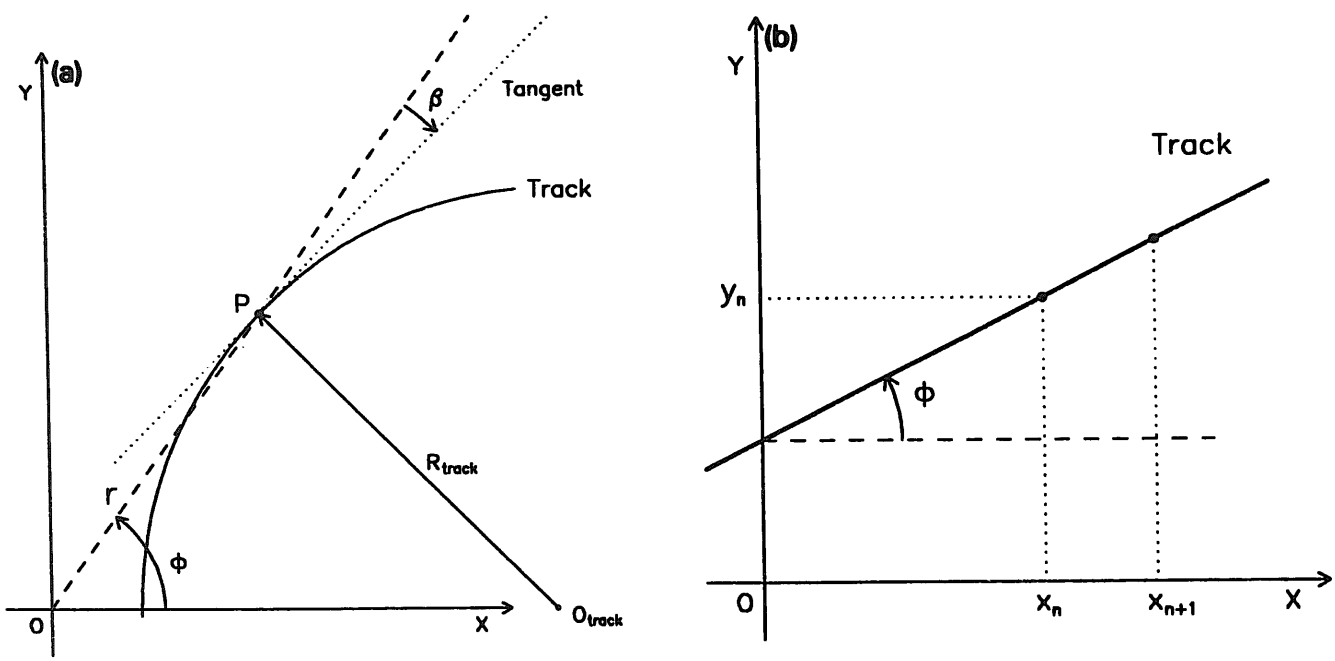

Fig. 3. Illustration of track parameters: (a) circular track; (b) linear track.

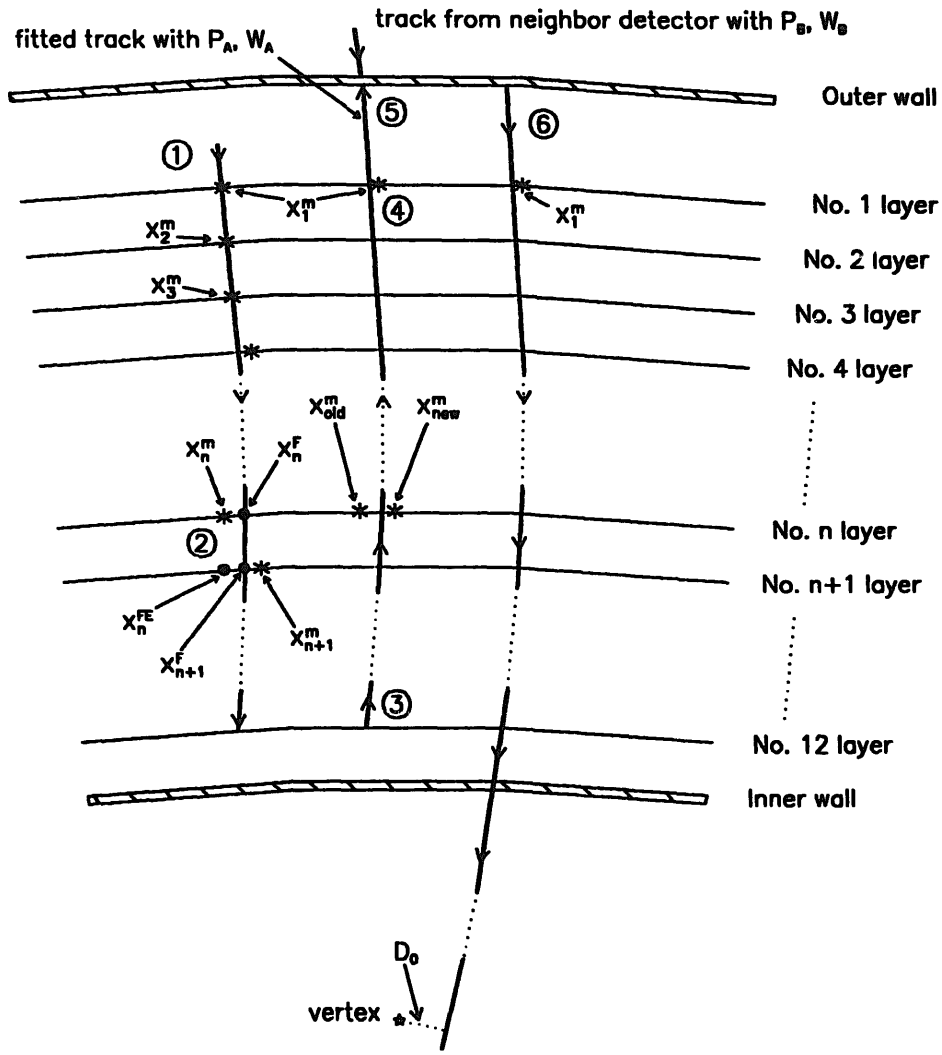

Fig. 4. Track-following procedures.

outermost (labelled as no. 1) wire layer as "starting" points. One starting point plus one compatible point on the lower layer forms a tentative segment. "Compatible" here means "within a window". The size of the window depends on the detector and has to be tuned: a large size allows to recognize low-p_t tracks (with large angle with respect to the radial direction) but it gives more combinations. For ZEUS VXD, the window size is chosen as the width of three cells, i.e. about 20 mm. Next, a segment plus another compatible point on a further lower layer forms a three-points chain, which can be exactly fitted as a circular track with track parameters P_3^F and associated weight matrix $\mathbf{W}_3$.

Another way to start tracks (not yet used in this study) is to follow the tracks from a neighbouring detector which carry P_0^F and $\mathbf{W}_0$. Since the information carried by those tracks generally comes from more than three points and thus is more accurate than the previous track-starting method, this may reduce the ambiguity introduced by the inaccuracy of starting tracks when we go to the next steps. However, some tracks may have been lost or badly measured in the neighbouring detector so that the previous starting method is always useful.

2.2. Following tracks

The procedures detailed below in steps 2.2.1 and 2.2.2 are repeated for all tracks, in parallel, until the last layer is reached.

2.2.1. Extrapolating tracks

If we already got the information of a candidate track on the nth layer, then the P_n^F and $\mathbf{W}_n$ are extrapolated to the next (i.e. $(n+1)$th) wire layer as

$P_n^F \Rightarrow P_n^{Fe}$, by a circle (for this example) or other parametrization extrapolation;

$$\mathbf{W}_n \Rightarrow \mathbf{W}_n^e = \mathbf{D}^T \mathbf{W}_n \mathbf{D}, \text{ where } \mathbf{D} = \partial P / \partial P^e. \quad (1)$$

For circle (two-dimensional) or helix (three-dimensional) parametrization, the matrix $\mathbf{D}$ can be calculated analytically [8]. If there is no magnetic field, parametrization and extrapolation are simply linear (see appendix A). For other cases, the extrapolation has to be computed precisely, since this is the first crucial step for the method.

2.2.2. Selecting a measured point x^m_{n+1} on the $(n+1)$th layer

Among all x^m_{n+1} within a window around the extrapolated point x^{Fe}_n on the $(n+1)$th layer, the closest point to the x^{Fe}_n is selected to be a candidate point. The window is narrower than in step 2.1, typically a few times the measurement error. If no point exists in the window, a missing point is counted and the track will be extrapolated to the next, i.e. $(n+2)$th, layer by step 2.2.1; else P^F_{n+1} is obtained by solving a system of linear equations which includes the information of x^{Fe}_n and x^m_{n+1}.

The system of equations is derived as follows. The total χ^2 due to the points 1 to $n+1$ can be expressed as

$$\chi^2_{(n+1)}(P_{n+1}) = \chi^2_{(n)}(P^F_n) + (P_{n+1} - P^{Fe}_n)^T \mathbf{W}^e_n (P_{n+1} - P^{Fe}_n) + (P_{n+1} - x^m_{n+1})^T \mathbf{U} (P_{n+1} - x^m_{n+1}),$$

where

$$\mathbf{U} = \begin{pmatrix} 1/\sigma^2_\phi & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (2)$$

Here σ_ϕ is the measurement error on the parameter ϕ. When P_{n+1} varies, we can say that the first two terms on the right side of eq. (2) are contributions of the points 1 to n, and the third term is contributed by the candidate point x^m_{n+1}.

Minimizing the χ^2 gives a system of linear equations

$$(\mathbf{W}^e_n + \mathbf{U})(P_{n+1} - P^{Fe}_n) = \mathbf{U} \begin{pmatrix} \phi^m_{n+1} - \phi^{Fe}_n \\ 0 \\ 0 \end{pmatrix}. \quad (3)$$

Its solution gives the fitted parameters P^F_{n+1} (the explicit expressions of eq. (3) and its solutions for the 2D straight-line case can be found in appendix A). The updated weight matrix is $\mathbf{W}_{n+1} = \mathbf{W}^e_n + \mathbf{U}$. Substituting the P^F_{n+1} back into eq. (2), we can get the updated global χ^2 for the track. The increase of χ^2 is $\delta\chi^2 = \chi^2_{(n+1)}(P^F_{n+1}) - \chi^2_{(n)}(P^F_n)$, i.e. it is the sum of the last two terms on the right side of eq. (2).

If the $\delta\chi^2$ is smaller than a given maximum-tolerance value, the candidate point is added to the track; otherwise a missing point is counted. If points are missing on a few (e.g. three) consecutive layers, the track is declared dead; otherwise the steps 2.2.1 and 2.2.2 will be repeated until the last (i.e. the 12th in ZEUS VXD) wire layer.

2.3. Backwards following to rescue the track

It happens sometimes that a candidate track dies in midway and restarts after its death, i.e. the track is split into two or more pieces. This reduces the track-finding efficiency significantly if the definition of the efficiency requires a high quality (i.e. a small number of wrong points) of reconstructed tracks and no double counting. To cope with this problem, the procedure of backwards following is necessary.

Going backwards, we start from P^F and $\mathbf{W}$ at the last point (say on the nth layer) of the candidate track, then extrapolate it to the $(n-1)$th layer. When the point selection as at the beginning of step 2.2.2 is done, one of the following three possibilities will occur:

(a) If a point was missing on this layer, the newly selected point is added under the condition of maximum allowed $\delta\chi^2$.

(b) If the new point is the same as the old one, nothing needs to be changed. The track is directly extrapolated to the next (i.e. $(n-2)$th) layer.

(c) If the new point is different from the old one, the old point is removed by solving eq. (3) with $-\mathbf{U}$ instead of $\mathbf{U}$. Then the new point is added on the track with eq. (3).

The process keeps going on to the first layer. In our experience, this step has recovered many missing points and relinked many split tracks, therefore has greatly improved the track-finding and point-finding efficiencies. Actually this step also displays an advantage over other pattern-recognition methods because it needs no new tool.

2.4. Closing the "open" tracks

The candidate tracks in all prior steps can be called "open" candidate tracks. After removing all ghost tracks and applying certain cuts which depend on detector type and will be explained in section 3.2 and fig. 5, we get the "closed" candidate tracks for the next step.

2.5. Linking to neighbour detectors [9]

At first, the closed tracks have to be extrapolated (for both P^F and $\mathbf{W}$) to a common matching border in between two detectors. The extrapolation is the same as step 2.2.1. If the tracks have to pass through a material at position x, the multiple scattering due to the material can be taken into account by replacing $\mathbf{W}$:

$$\mathbf{W}_{new}(x) = \left(\mathbf{W}^{-1}_{old}(x) + \Sigma_{m/s}\right)^{-1},$$

where $\Sigma_{m/s}$ is the error matrix due to the multiple scattering. Then the extrapolation can be continuously

carried on from x to the next piece of material or to the common border. After tracks in both detectors (say detectors A and B) arrive at the border with P_A, P_B, W_A and W_B, the matching compatibility, e.g. the χ^2-like quantity $(P_A - P_B)^T (W_A^{-1} + W_B^{-1})^{-1} (P_A - P_B)$, can be tested within a suitable window. Once this χ^2-like quantity is smaller than a tolerance value, the final track parameter P_f at the border can be obtained by solving the equation $(W_A + W_B)P_f = W_A P_A + W_B P_B$; the associated weight matrix W_f is simply $W_A + W_B$.

2.6 Computing the impact parameter D_0 and its error

If there is no measurement point between the vertex and the innermost points of the detector, the candidate tracks can be extrapolated towards the vertex for getting D_0. The track is first extended to the inner wall of the detector, then the multiple scattering of the wall is included into the weight matrix, then the extrapolation is continued to the vertex. D_0 is the closest distance between the track and the vertex in the x–y plane.

3. Performance

3.1. Preparation of input

For testing the method, the simulated Monte Carlo LUND e–p events at HERA energy (30 GeV(electron) + 800 GeV(proton)) are used [10] and traced with GEANT [11] in VXD. The simulated tracks are digitized according to the sense wire positions and electronics limitations. For hardware reasons, the positions of sense wires can not be staggered in VXD, so that the left–right (L–R) ambiguity can not be solved until a late stage of track reconstruction. In other words, each digitization will create two (one real and one image) digitized points. They are smeared in terms of the detector's resolution (about 55 μm for the points closer than 1.2 mm to the sense wires, about 35 μm for other points) which is determined by experimental beam tests of the detector prototype. The input of the reconstruction program are these smeared points which are well

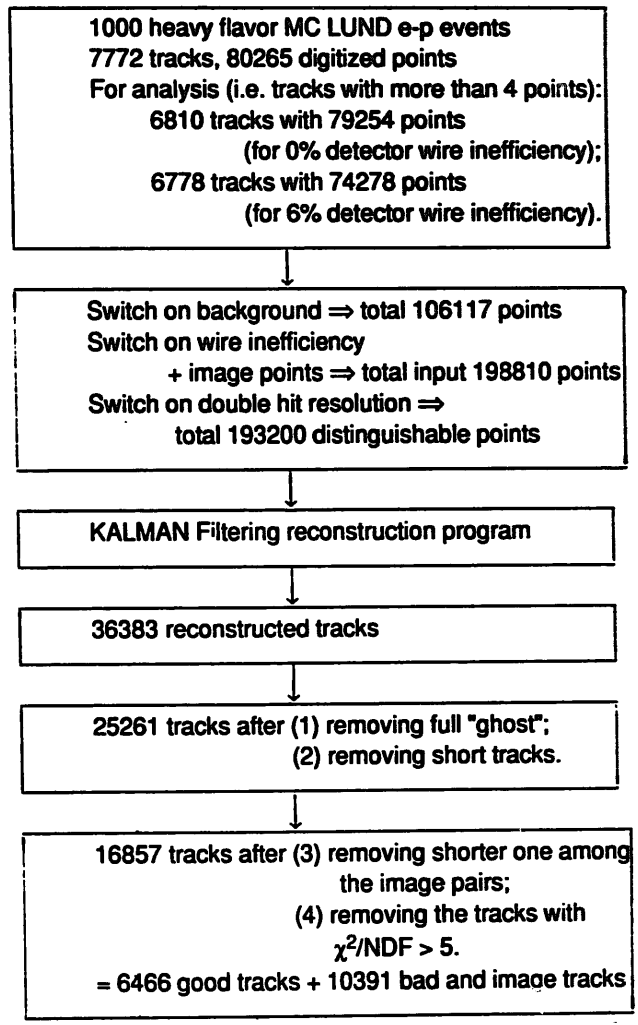

Fig. 5. Flowchart of result analysis.

Table 1
Result and statistics of the test

Input options	background	off	on	on	on	on	on
	limited double hit resolution	off	off	on	off	on	on
	wire inefficiency (%)	0	0	0	6	6	4
ϵ_{trk} (%)		98.1	97.9	97.0	96.7	95.4	95.7
$\epsilon_{pnt}(1)$ (%)		97.6	97.4	96.2	96.2	94.7	95.0
$\epsilon_{pnt}(2)$ (%)		99.4	99.3	99.0	99.2	98.9	98.9
Total number of candidate reconstructed tracks		12878	18342	18109	17081	16857	17172
Total used tracks		6810	6810	6810	6778	6778	6789
Total used points		79254	79254	79254	74278	74278	76020
Total digitizations (when including background)		80265	106117	106117	106117	106117	106117
Total input points (when including wire inefficiency)		160530	211658	211658	198810	198810	203132
Total distinguishable points (when including double-hit resolution)		160530	211658	205352	198810	193200	197282

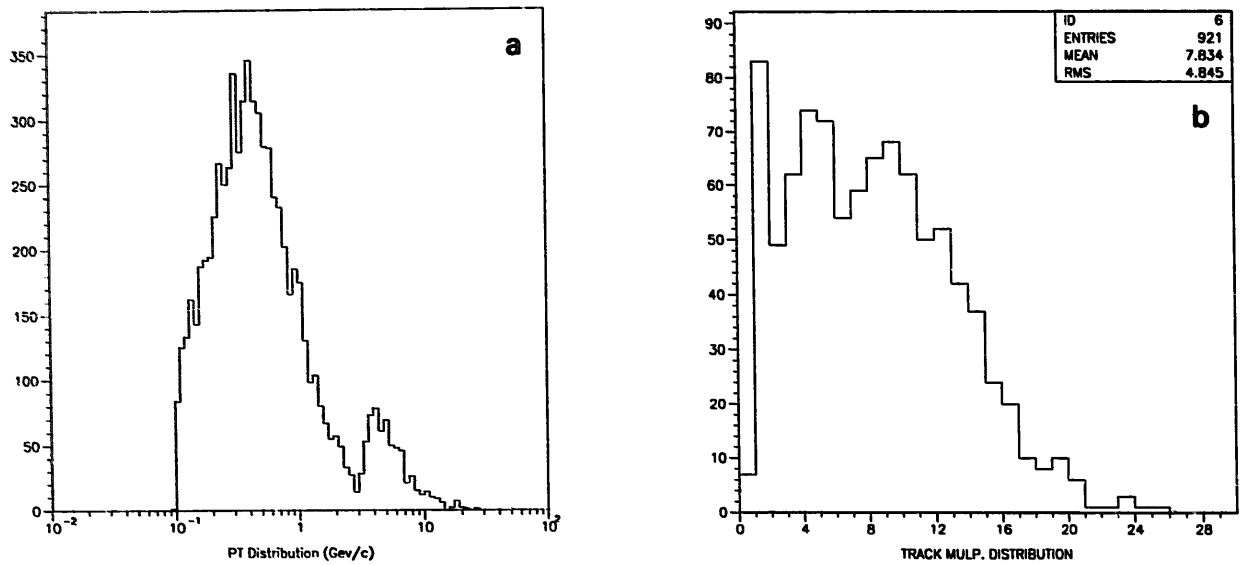

Fig. 6. Transverse momentum p_t and track multiplicity distributions of the 1000 simulated heavy flavour LUND events.

labelled so that the output of the program can be judged by identifying the index attached on each point.

In order to be as realistic as possible, three other side-effects are simulated also, as follows:

(a) Background events which include beam–gas scattering (proton–proton LUND events), beam halo and synchrotron radiation [10]. Since VXD is one of the detectors closest to the beam, background is rather serious; from table 1, it can be seen that the background events can product about 30% more input points to confuse the event coming from the vertex.

(b) Detector wire inefficiency of 6 or 4%. I.e. at that percentage of probability a sense wire of the detector does not get any signal. The possible 4% wire inefficiency is obtained in a recent beam test.

(c) Double-hit resolution due to the limitation of electronics. I.e. if two points are too close to each other, electronics will not be able to distinguish them. In real experiment, we will only see the first arriving signal with a broad width instead of two separated narrow signals; this can be simulated by moving the second point into the position of the first one if the distance between two points is smaller than a minimum distinguishable value in a cell. At the same time, the measured errors for both points are enlarged as well. From table 1 it can be seen that the limited double-hit resolution reduces the distinguishable points for about 3%.

These side-effects are switched on one by one in order to see how each item influences the results.

3.2. Analysis procedures

The output of the Kalman filtering program undergoes the following analyzing procedures.

(a) Remove the full "ghost" tracks (which means that the points on a track are exactly the same as the points on another track).

(b) Remove short tracks with four points or less. The short tracks will be handled after the first-round track finding.

(c) Group the real and image track pairs. If the lengths (i.e. the number of points) of tracks in the pair are the same or differ by only one point, both tracks will be kept and go to the next step; otherwise, the shorter one will be removed.

(d) Remove tracks with $\chi^2/\mathrm{NDF} > 5$, where NDF is the number of degrees of freedom. But if χ^2/NDF of both tracks in the image pair are larger than 5, both will be kept unless one of them is extremely odd (e.g. with very high χ^2).

The procedures and sample numbers are illustrated in fig. 5.

3.3. Definitions of track-finding efficiency and point-finding efficiency

Among the final candidate tracks, we identify the index assigned to each point, and take out the whole group of image tracks. Then the efficiencies are calculated among all real tracks.

A track is considered as a good one only when less than 20% of its points are "wrong" by comparing it with the simulated track. One point is counted as "wrong" if it does not belong to the simulated track or if it is missing. Thus the track efficiency ϵ_{trk} is defined as

$$\epsilon_{\mathrm{trk}} \equiv 1 - \frac{\text{total number of (bad tracks + missing tracks)}}{\text{total number of simulated tracks used}}.$$

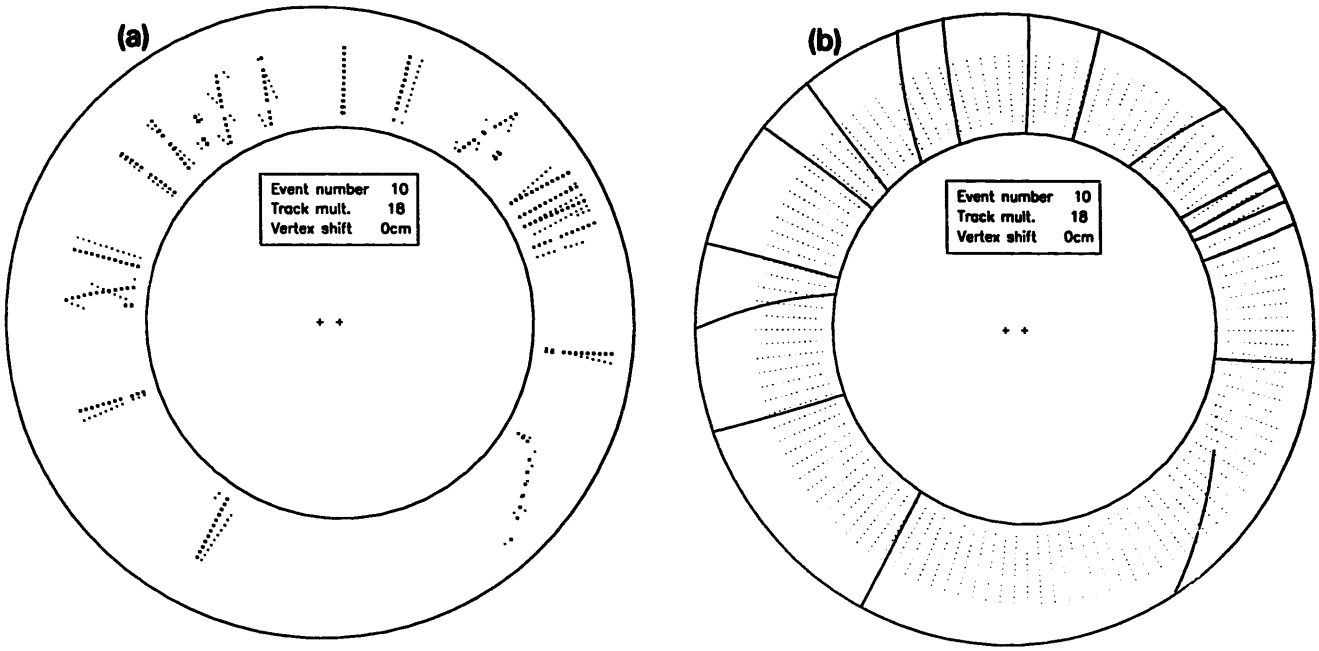

Fig. 7. Graphic display for a reconstructed event. (a) input points: dark dots = real points; light dots = image points; crosses = backgrounds; plusses = beam-pipe centre and beam positions. (b) reconstructed event: solid lines = reconstructed tracks; dotted lines = cell walls.

A simulated track is used for analysis only if it has more than four points.

Two point-finding efficiencies can be defined as:

$$\epsilon_{\mathrm{pnt}}(1) \equiv 1 - \frac{\text{total number of ("wrong" points on all good tracks + all points on all bad and missing tracks)}}{\text{total number of points on all simulated tracks used}},$$

$$\epsilon_{\mathrm{pnt}}(2) \equiv 1 - \frac{\text{total number of "wrong" points on all good tracks}}{\text{total number of points on all good tracks}}.$$

Obviously, $\epsilon_{\mathrm{pnt}}(1)$ is related with ϵ_{trk}, and is always lower.

3.4. Results

(a) When the primary vertex of events is at the origin, with 7772 tracks and 80265 points of 1000 simulated neutral current events ($Q^2 \geq 10$ GeV^2) including the production of heavy flavours (c, d and t), table 1 shows the result and statistics. The input tracks have transverse momentum spread from 0.08 to 26.4 GeV/c and peaked at 0.4 GeV/c (fig. 6a). Charged-track multiplicities of events range from 0 to 25, with a mean value of 7.8 (fig. 6b). The secondary vertices are located within a few mm (for decays of heavy flavour particles) or up to the order of 10 cm (for K^0 and Λ decays etc.).

From table 1, it can be seen that under about 30% background contamination and 4% wire inefficiency, the track-finding efficiency remains at 95.7%, and the point-finding efficiency is near 99%. The total number of candidate reconstructed tracks is more than the simulated tracks by a factor 2 to 3, due to the unsolved L–R ambiguity. The problem will be untangled after the track matching with neighbour detectors.

(b) Fig. 7 is a graphic display for a typical event which involves a kaon K^+ decay to π^+ and some neutral particles.

(c) Figs. 8 and 9 show the track-fitting qualities which are obtained through steps 2.1 to 2.4 of section 2. For each track parameter P, the quantities ($P_{\mathrm{fitted}} - P_{\mathrm{generated}}$) and ($P_{\mathrm{fitted}} - P_{\mathrm{generated}}$)/$\sigma_P$ are histogrammed, where σ_P is the square root of the corresponding diagonal element in the error matrix.

These plots demonstrate that the method is quite satisfactory as a track fitter. Therefore, together with the track-finding qualities shown above we can claim the simultaneity of pattern recognition and track fitting.

Of course, with only 12 points, track parameters can not be determined accurately, by whatever track-fitting method. The precise parameters will be obtained after the track matching as step 2.5 of section 2. Before the matching, each detector has to fit tracks by its own; the results prove that the Kalman filtering is competent to do such a job.

4. Flexibility

The Kalman filtering method is quite flexible to be applied in various situations, such as:

1) It can be used as a major track finder and fitter, this is demonstrated in the above sections.

2) It can also be used as a supplementary track finder and fitter. After first-round pattern recognition and track fitting for each adjacent detector independently, after track matching between the adjacent detectors, the method can pick up unused points in a detector to find and fit additional tracks by following the tracks from neighbour detectors which have no match with a track in this detector.

3) It still can be used as a track fitter only, by using the points found instead of the points selected in steps 2.1 and 2.2 of section 2. It is particularly powerful to find kinks in a track. This capacity is not given by some other fitters (e.g. global fitting methods). More details on the kink finding can be found in ref. [12].

The above flexibility can be applied separately: if there would exist another more powerful major track

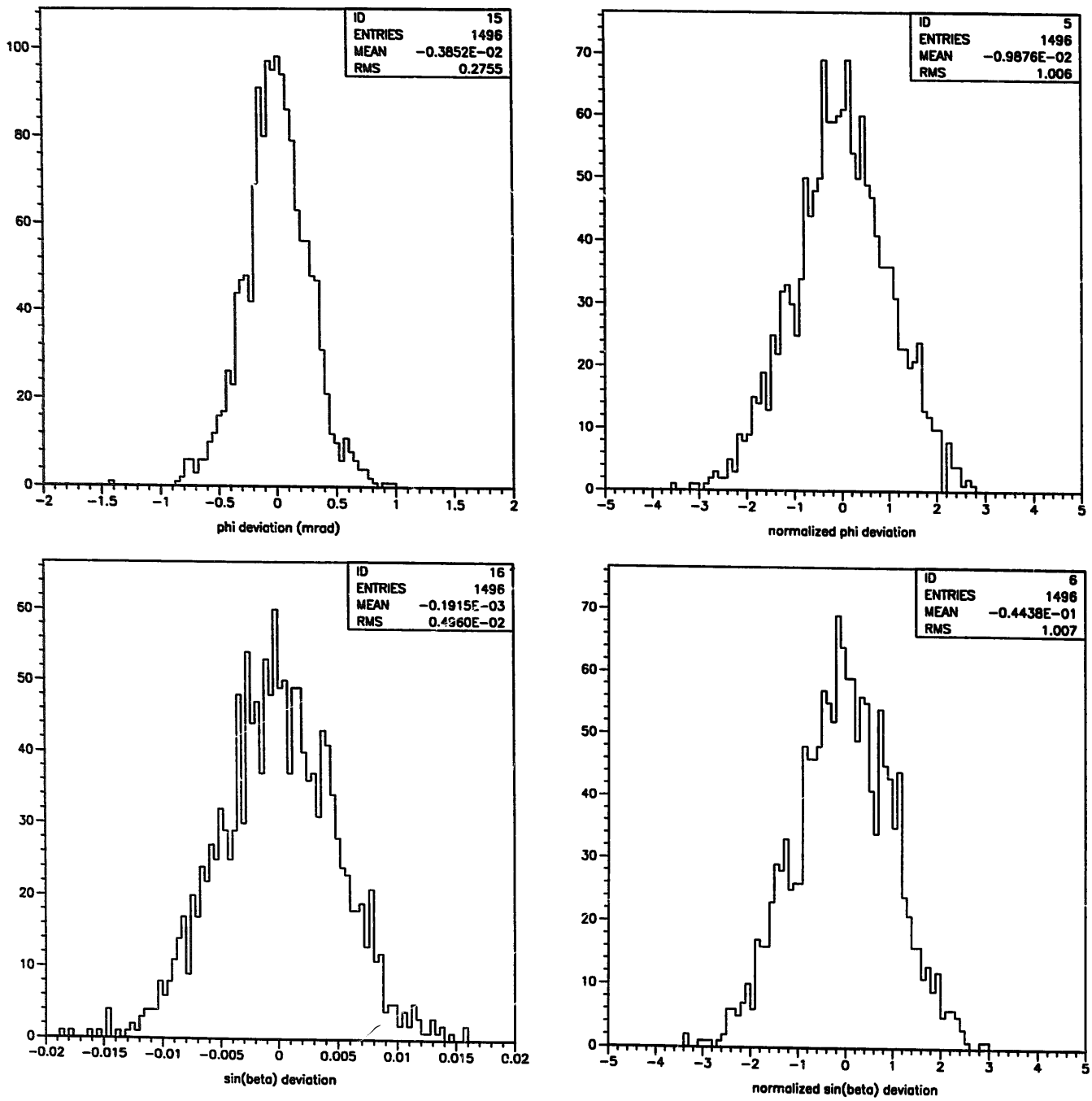

Fig. 8. Deviations and normalized-deviations plots for three track parameters.

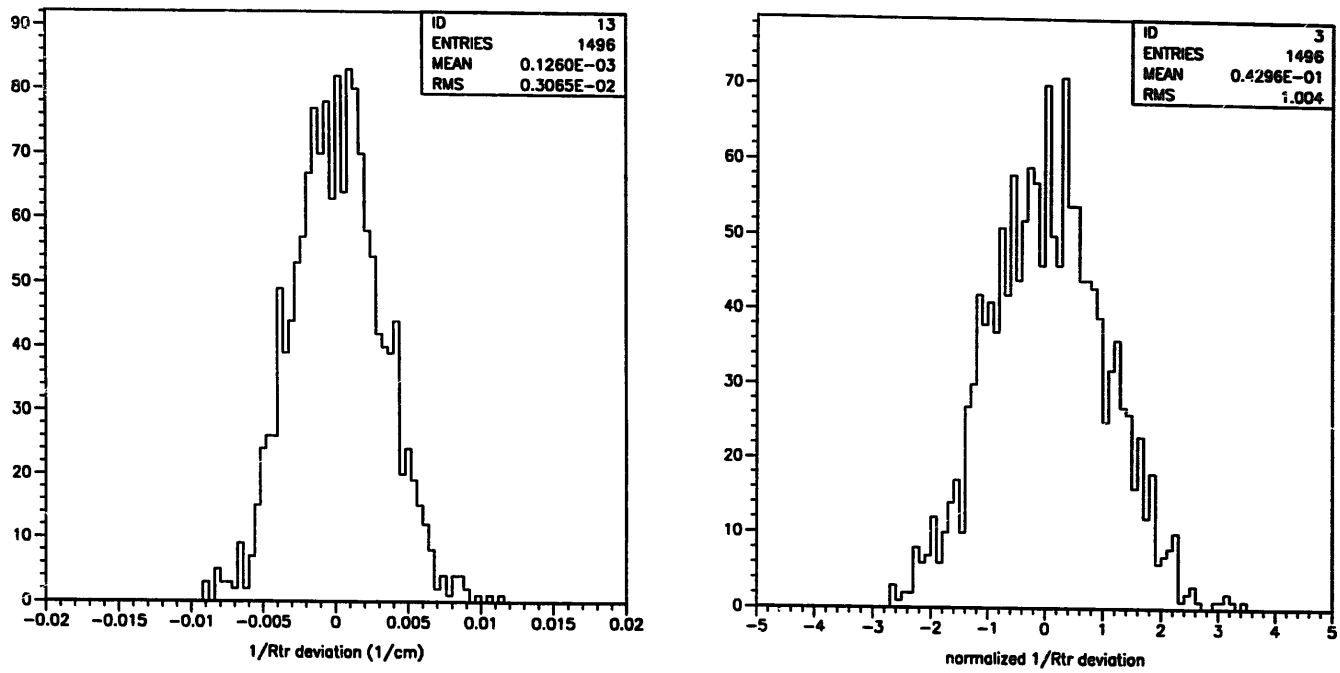

Fig. 8 (continued).

finder, then the Kalman filtering method can play a role in points (2) and/or (3) above; if there would exist another better way to do jobs (1) and (2), the Kalman filtering still can do job (3). Of course, it is preferable to let one tool accomplish all three jobs if it is qualified to do so. Our practice has shown that the Kalman filtering method is a strong candidate to be such a tool.

5. Conclusions

The Kalman filtering method is working very well in our practice.

For the pattern-recognition part,

- the track-finding efficiency can reach 95.7% or more under heavy (about 30%) background contamination;

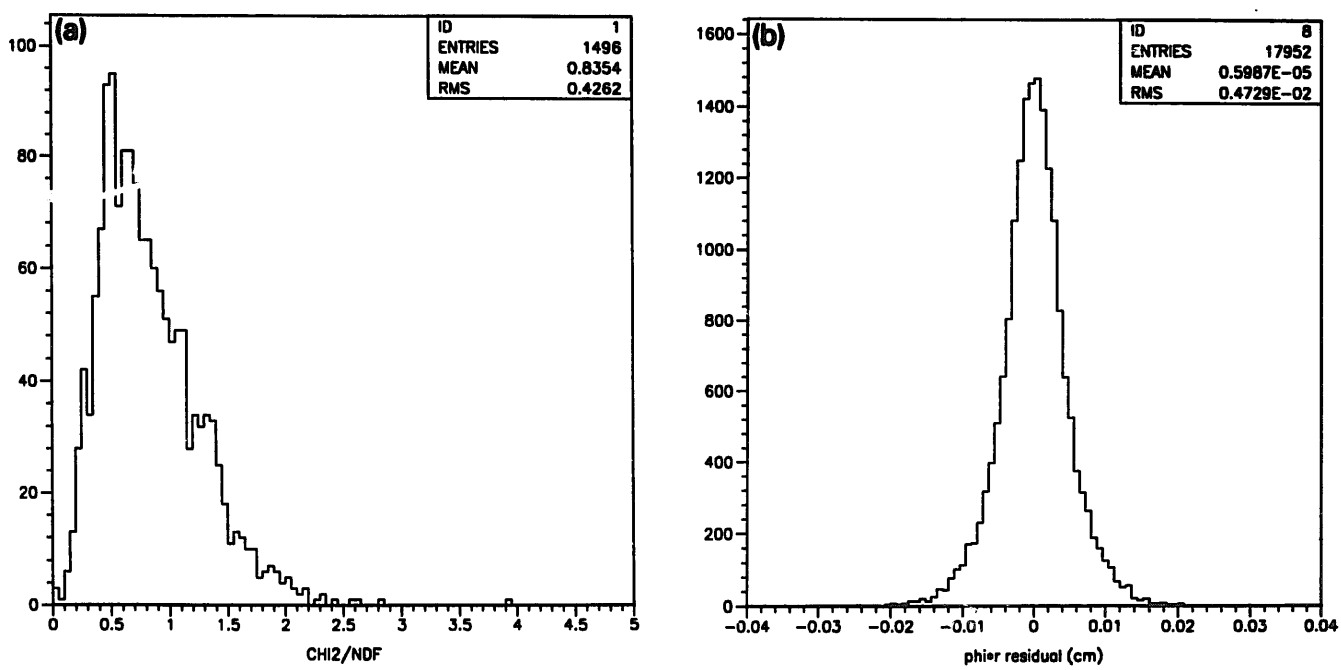

Fig. 9. Track fitting qualities: distribution of (a) χ^2/NDF, (b) residuals.

– the point-finding efficiency is near and greater than 99%.

For the track-fitting part, all recognized tracks are fitted well without any separate fitting program.

The major advantages are that it can fulfil two tasks (i.e. pattern recognition and track fitting) simultaneously and it is quite flexible to play roles in various situations without introducing new tools or new programs. Therefore the Kalman filtering can provide an efficient way for track reconstruction.

Further studies are in progress in order to evaluate the dependence of track-finding qualities on vertex displacement, charged-track multiplicity and transverse momentum.

Acknowledgements

S. Qian is grateful to all members of the ZEUS VXD group, especially to Dr. G. Iacobucci who has generated heavy-flavour and background events for testing this and other methods, and to Dr. S. D'Auria for the help on graphic display. He also thanks Dr. R. Nania for the discussion on result analysis. The work has been supported by the LAA group at CERN led by Prof. A. Zichichi and INFN/Frascati of Italy, we would like to express our sincere gratitude to them.

Appendix A

A straight line can be parametrized as $P_n = (y_n, A_n)|_{x_n}$ (see fig. 3b), where $A_n = \tan\phi$ at x_n. The measurement plane is parallel to the y-axis with error σ at fixed x_n.

The extrapolation:

$$P_n^{\mathrm{F}} \Rightarrow P_n^{\mathrm{Fe}} = \begin{pmatrix} y_n^{\mathrm{Fe}} \\ A_n^{\mathrm{Fe}} \end{pmatrix}\Bigg|_{x_{n+1}} = \begin{pmatrix} y_n^{\mathrm{F}} + \Delta x A_n^{\mathrm{F}} \\ A_n^{\mathrm{F}} \end{pmatrix}\Bigg|_{x_{n+1}},$$

where $\Delta x = x_{n+1} - x_n$;

$$\mathbf{W}_n \Rightarrow \mathbf{W}_n^{\mathrm{e}} = \mathbf{D}^{\mathrm{T}}\mathbf{W}_n\mathbf{D},$$

where

$$\mathbf{D} = \partial P/\partial P^{\mathrm{e}} = \begin{pmatrix} 1 & -\Delta x \\ 0 & 1 \end{pmatrix},$$

$$\therefore \mathbf{D}^{-1} = \begin{pmatrix} 1 & \Delta x \\ 0 & 1 \end{pmatrix}.$$

By denoting $\mathbf{W}_n$ as

$$\mathbf{W}_n = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix},$$

and noticing $\mathbf{W}_n$ is symmetric, i.e. $w_{21} = w_{12}$, we get

$$\mathbf{W}_n^{\mathrm{e}} = \begin{pmatrix} 1 & 0 \\ -\Delta x & 1 \end{pmatrix}\begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}\begin{pmatrix} 1 & -\Delta x \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} w_{11} & w_{12} - w_{11}\Delta x \\ w_{12} - w_{11}\Delta x & w_{22} - 2w_{12}\Delta x + w_{11}(\Delta x)^2 \end{pmatrix}.$$

The system equation is

$$(\mathbf{W}_n^{\mathrm{e}} + \mathbf{U})(P_{n+1} - P_n^{\mathrm{Fe}}) = \mathbf{U}\begin{pmatrix} y_{n+1}^{\mathrm{m}} - y_n^{\mathrm{Fe}} \\ 0 \end{pmatrix},$$

where

$$\mathbf{U} = \begin{pmatrix} 1/\sigma^2 & 0 \\ 0 & 0 \end{pmatrix}.$$

By denoting $\mathbf{W}_n^{\mathrm{e}}$ as

$$\mathbf{W}_n^{\mathrm{e}} = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix},$$

the above equation can be explicitly expressed as

$$\begin{pmatrix} V_{11} + (1/\sigma^2) & V_{12} \\ V_{21} & V_{22} \end{pmatrix}\begin{pmatrix} y_{n+1}^{\mathrm{F}} - y_n^{\mathrm{Fe}} \\ A_{n+1}^{\mathrm{F}} - A_n^{\mathrm{Fe}} \end{pmatrix} = \begin{pmatrix} (y_{n+1}^{\mathrm{m}} - y_n^{\mathrm{Fe}})/\sigma^2 \\ 0 \end{pmatrix}.$$

Its solutions are (again noticing $V_{21} = V_{12}$):

$$y_{n+1}^{\mathrm{F}} = \left[(y_{n+1}^{\mathrm{m}} - y_n^{\mathrm{Fe}})/\sigma^2\right] / \left[V_{11} + (1/\sigma^2) - (V_{12})^2/V_{22}\right] + y_n^{\mathrm{Fe}},$$

$$A_{n+1}^{\mathrm{F}} = -(y_{n+1}^{\mathrm{m}} - y_n^{\mathrm{Fe}})V_{12}/V_{22} + A_n^{\mathrm{Fe}}.$$

From y_{n+1}^{F} and A_{n+1}^{F}, the increment $\delta\chi^2$ can be computed by eq. (2) of section 2.2.2.

References

[1] R.E. Kalman, J. Basic Eng. 82 (1961) 34;
R.E. Kalman and R.S. Bucy, ibid., 83 (1961) 95.
[2] R. Fruhwirth, Nucl. Instr. and Meth. A262 (1987) 444.
[3] P. Billoir, Comput. Phys. Commun. 57 (1989) 390.
[4] P. Billoir, Nucl. Instr. and Meth. 225 (1984) 352.
[5] D. Savard et al., Nucl. Instr. and Meth. A268 (1988) 278.
[6] The Kalman Filtering method may be expressed in two mathematically equivalent formalisms. The original one [1] uses the covariance (i.e. error) matrix throughout the computation; the other one [4] uses its inverse (i.e. the weight matrix). The choice of formalism depends on convenience.
[7] ZEUS Collaboration, The ZEUS detector status report 1989, PRC-89/01, (March 1989).
[8] P. Billoir, DELPHI 86-66 (1986);
ibid. 87-4 (1987).
[9] P. Billoir, R. Fruhwirth and M. Regler, Nucl. Instr. and Meth. A241 (1985) 115.
[10] G. Iacobucci, Double hit resolution in ZEUS VXD, Z-2 internal note (May 1989).
[11] R. Brun et al. GEANT3, CERN DD/EE/84-1 (1986).
[12] M. Regler and R. Fruhwirth, Proc. Advanced Study Institute on Techniques and Concepts in High Energy Physics, St. Croix, Virgin Islands (Plenum, 1989).