ELSEVIER

Computer Physics Communications 100 (1997) 1–16

Computer Physics Communications

Track fitting with non-Gaussian noise

R. Frühwirth [1]

Institut für Hochenergiephysik der Österreichischen Akademie der Wissenschaften, Nikolsdorfer Gasse 18, A-1050 Vienna, Austria

Received 30 September 1996

Abstract

This paper is a continuation of a previous note on robust track fitting. The nonlinear filter which has been investigated there is extended by running several Kalman filters in parallel. The number of filters is kept small by a suitable collapsing procedure. We report results on the precision of the estimated track parameters and on the computational load required by the algorithm with non-Gaussian observation errors. We also develop the correct smoothing algorithm and discuss the treatment of non-Gaussian process noise.

1. Introduction

In a previous publication [1] we have investigated a nonlinear robust filter derived by Bayesian principles [2]. We have shown that it produces estimates with a smaller variance than does the optimal linear filter (Kalman filter), if the observation error is contaminated with outliers or tails. The construction of the filter is based on a Gaussian mixture model of the observation error. The first component of the mixture describes the regular measurements or the core of the distribution, the second one describes the outliers or the tails.

The robust filter works in a way similar to the Kalman filter, by alternating prediction and filter steps. If the distribution of the predicted state vector (the "prior") is Gaussian, the distribution of the filtered state vector (the "posterior") is a mixture of two Gaussians. An exact prediction in the subsequent filter step would result in the posterior being a mixture of four Gaussians, and continuing in this way would yield an exponentially increasing number of components. In order to keep the algorithm simple, the posterior distribution is "collapsed" after each filter step, i.e. it is replaced by a single Gaussian with the mean vector and the covariance matrix of the mixture. Thus the prior is a Gaussian distribution in all filter steps.

This robust filter was designed to cope with Gaussian noise contaminated by outliers or tails. In this note we investigate a more general procedure which can be applied to a broader class of non-Gaussian distributions, by allowing all densities involved to be mixtures of several Gaussians. The resulting algorithm, in which several Kalman filters run in parallel, is also known as the Gaussian-sum filter [3]. Again, we cannot tolerate an exponentially increasing number of components; we therefore have to find a way of limiting the number of

[1] E-mail: fruhwirth@hephy.oeaw.ac.at

PII S0010-4655(96)00155-5

components of the posterior after each filter step. An obvious, though not necessarily the best solution to this problem, is the suppression of those components in the Gaussian mixture which have the smallest weights. Alternatively, we may try to combine components which are in some sense close to each other to a single component. As the computational load is directly proportional to the number of components, a balance has to be found between the quality of the approximation to the current posterior and the speed of the algorithm.

We see several applications for the parallel filter:

- It is just a matter of convenience that in the robust filter mentioned above the posterior is approximated by a single Gaussian. With the parallel filter we can use a mixture of as many Gaussians as we wish. We will show below that the performance of the robust filter can be further improved in this way.
- The same is true if the process noise is described by a Gaussian mixture model. This is particularly pertinent to multiple scattering in an inhomogeneous material. In this case it is the usual practice to "smear" the material and to work with an average thickness and an average radiation length. With the parallel filter it is now possible to describe the distribution of multiple scattering in greater detail without forfeiting the computational simplicity of the Kalman filter.
- The parallel filter is able to handle cases where both process noise and observation error are Gaussian mixtures.
- Finally, the parallel filter can be applied to more general distributions than the two-component outlier model, as long as a good approximation as a Gaussian mixture is available. For instance, it is now feasible to treat energy loss of electrons or muons more precisely by setting up a Gaussian mixture model of the actual distribution.

In Section 2 we work out the mathematical details of the parallel filter in the context of track fitting. In Section 3 its performance is evaluated under the assumption of a Gaussian mixture distribution of the observation error. In particular, we report on the variance of the estimate and the computational load compared to the optimal linear filter. In Section 4 we give the correct prescription for smoothing with the parallel filter. We conclude with some hints on the treatment of non-Gaussian process noise (Section 5) and on the efficient implementation of the algorithm (Section 6).

2. The parallel filter

We start by recalling the linear track model suitable for estimation of the track parameters by the Kalman filter [4]. In most cases the linear track model is actually a first order Taylor approximation to a nonlinear model, which is in turn a solution of the equation of motion. The model is specified by a set of system equations and a set of measurement equations.

System equations:

$$\boldsymbol{x}_k = \boldsymbol{F}_k \boldsymbol{x}_{k-1} + \boldsymbol{c}_k + \boldsymbol{\omega}_k ,$$
$$\mathrm{E}(\boldsymbol{\omega}_k) = 0 , \quad \mathrm{cov}(\boldsymbol{\omega}_k) = \boldsymbol{Q}_k , \quad k = 1, \ldots, n .$$

Measurement equations:

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{d}_k + \boldsymbol{\epsilon}_k ,$$
$$\mathrm{E}(\boldsymbol{\epsilon}_k) = 0 , \quad \mathrm{cov}(\boldsymbol{\epsilon}_k) = \boldsymbol{V}_k = \boldsymbol{G}_k^{-1} , \quad k = 1, \ldots, n .$$

It is assumed that the track detector can be represented as a collection of shells or surfaces, each of which contributes a measurement. $\boldsymbol{x}_k$ denotes the state vector of the five track parameters at measurement surface k, i.e. the intersection point, the track direction, and the curvature. In general, the state vector cannot be observed

directly. The evolution of the state vector is described by the system matrix F_k and the constant term c_k. The process noise between surface $k-1$ and k is denoted by ω_k. If energy loss is neglected, it is the sum of the integrated continuous multiple scattering plus all discrete scattering between surface $k-1$ and k. The measurements in surface k are denoted by y_k, and the associated observation error by ϵ_k. The linear function which maps the state vector x_k on the measurement vector y_k is defined by the matrix H_k and the constant term d_k. Without loss of generality we may assume that all c_k and all d_k are equal to zero.

Recursive least-squares estimates of the state vector are provided by the Kalman filter which is the optimal linear filter. The estimation formulas are well known and need not be spelled out here. They can be found in the literature [4], along with the formulas for the covariance matrices of the estimated state vectors and the associated χ^2-statistics.

In the most general case, both process noise and observation errors are modeled by a mixture of Gaussian densities. Let us consider the filter step at measurement surface k. The predicted distribution of the state vector x_k conditional on the observations $Y_{k-1} = \{y_1, \ldots, y_{k-1}\}$ can be assumed to be a Gaussian mixture with N_{k-1} components,

$$p(x_k|Y_{k-1}) = \sum_{j=1}^{N_{k-1}} \pi_k^j \varphi(x_k; x_{k|k-1}^j, C_{k|k-1}^j) , \quad \sum_{j=1}^{N_{k-1}} \pi_k^j = 1 ,$$

where $\varphi(\cdot; \mu, V)$ is a multivariate Gaussian p.d.f. with mean μ and covariance matrix V. This is the prior distribution of the state vector x_k.

We further assume that the distribution of the observation error ϵ_k can be modeled by the following Gaussian mixture with M_k components:

$$p(\epsilon_k) = \sum_{i=1}^{M_k} p_k^i \varphi(\epsilon_k; 0, V_k^i) , \quad \sum_{i=1}^{M_k} p_k^i = 1 .$$

Therefore the p.d.f. of the observation y_k conditional on the state x_k is given by

$$p(y_k|x_k) = \sum_{i=1}^{M_k} p_k^i \varphi(y_k; H_k x_k, V_k^i) .$$

Application of Bayes' theorem then leads to the following posterior distribution:

$$p(x_k|Y_k) \equiv p(x_k|y_k, Y_{k-1}) = \frac{p(y_k|x_k) p(x_k|Y_{k-1})}{\int p(y_k|x_k) p(x_k|Y_{k-1}) \, dx_k} .$$

By Lemma 2 of the appendix, the posterior can be written in the following way:

$$p(x_k|Y_k) = \sum_{i=1}^{M_k} \sum_{j=1}^{N_{k-1}} q_k^{ij} \varphi(x_k; x_{k|k}^{ij}, C_{k|k}^{ij}) .$$

Thus the posterior is a mixture of $n_k = M_k N_{k-1}$ Gaussian components with the following posterior weights:

$$q_k^{ij} \propto p_k^i \pi_k^j \varphi(y_k; H_k x_{k|k-1}^j, V_k^i + H_k C_{k|k-1}^j H_k^T) .$$

The posterior weights are functions of the observation and of the prediction. The constant of proportionality is determined by the requirement that the sum of all q_k^{ij} is equal to 1. The mean and the covariance matrix of each component is obtained by a Kalman filter,

$$\boldsymbol{x}_{k|k}^{ij} = \boldsymbol{x}_{k|k-1}^{j} + \boldsymbol{C}_{k|k}^{ij}\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{G}_k^{i}(\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{x}_{k|k-1}^{j}) ,$$

$$\boldsymbol{C}_{k|k}^{ij} = \left[(\boldsymbol{C}_{k|k-1}^{j})^{-1} + \boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{G}_k^{i}\boldsymbol{H}_k\right]^{-1} .$$

There is a "chi-square" statistic associated to each Kalman filter,

$$(\chi_F^2)_k^{ij} = (\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{x}_{k|k}^{ij})^{\mathrm{T}}(\boldsymbol{V}_k^{i} - \boldsymbol{H}_k\boldsymbol{C}_{k|k}^{ij}\boldsymbol{H}_k^{\mathrm{T}})^{-1}(\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{x}_{k|k}^{ij}).$$

The final Bayesian estimate $\boldsymbol{x}_{k|k}$ and its covariance matrix $\boldsymbol{C}_{k|k}$ are obtained as the mean and the covariance matrix of the posterior distribution $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$. After renumbering the components with a single index, the posterior can be written as

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) = \sum_{l=1}^{n_k} q_k^l \varphi(\boldsymbol{x}_k; \boldsymbol{x}_{k|k}^l, \boldsymbol{C}_{k|k}^l) .$$

Then the mean and the covariance matrix of the mixture are equal to

$$\boldsymbol{x}_{k|k} = \sum_{l=1}^{n_k} q_k^l \boldsymbol{x}_{k|k}^l ,$$

$$\boldsymbol{C}_{k|k} = \sum_{l=1}^{n_k} q_k^l \boldsymbol{C}_{k|k}^l + \sum_{l=1}^{n_k}\sum_{m>l} q_k^l q_k^m (\boldsymbol{x}_{k|k}^l - \boldsymbol{x}_{k|k}^m)(\boldsymbol{x}_{k|k}^l - \boldsymbol{x}_{k|k}^m)^{\mathrm{T}} .$$

We also compute the "chi-square" statistic of the filter step as the appropriate weighted sum,

$$(\chi_F^2)_k = \sum_{l=1}^{n_k} q_k^l (\chi_F^2)_k^l .$$

By summing all "chi-squares" we obtain a "total chi-square" statistic of the track. It is, of course, not actually chi-square distributed.

After every filter step the posterior distribution has M_k times as many components as has the prior one. If the process noise in the subsequent prediction step is modeled by a Gaussian mixture as well, the number of components N_k in the next prior is again larger than the current number n_k of posterior components (see Section 5). Since the number of components thus rises very rapidly, we have to find a way of reducing the number of components in the posterior. Let us assume that the maximum number of components allowed in the posterior is equal to M. It is tempting to keep just the M components with the largest posterior weights and to drop the rest. This will be referred to as Algorithm D. Unfortunately, it turns out that the remaining mixture is but a poor approximation of the exact posterior. We shall see that it is much better, albeit slower, to cluster components which have small symmetric Kullback–Leibler distance, as proposed in [3]. In general, the symmetric Kullback–Leibler distance D of two probability density functions p_1 and p_2 is defined by

$$D(p_1, p_2) = 2(d(p_1, p_2) + d(p_2, p_1)) = 2 \cdot \left(\int \log \frac{p_1}{p_2}\, p_1\, dx + \int \log \frac{p_2}{p_1}\, p_2\, dx \right) .$$

If p_i is a Gaussian p.d.f. with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{V}_i = \boldsymbol{G}_i^{-1}$, the distance is equal to

$$D(p_1, p_2) = \mathrm{tr}[(\boldsymbol{V}_1 - \boldsymbol{V}_2)(\boldsymbol{G}_2 - \boldsymbol{G}_1)] + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}}(\boldsymbol{G}_1 + \boldsymbol{G}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) .$$

In the simple example studied below, the posterior distribution has twice as many components as the prior one. In this case, clustering is tantamount to collapsing pairs of distribution close to each other. The collapsing

algorithm, referred to below as Algorithm C, starts with the component with the largest weight, say component l with weight q_k^l. We find the component with the smallest distance D, say component m with weight q_k^m. These two are then collapsed to a single component which is a Gaussian p.d.f. $\varphi(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{V})$ with the same mean and covariance matrix as the mixture of components l and m,

$$\boldsymbol{\mu} = \frac{1}{q_k^l + q_k^m}(q_k^l \boldsymbol{x}_{k|k}^l + q_k^m \boldsymbol{x}_{k|k}^m) ,$$

$$\boldsymbol{V} = \frac{1}{q_k^l + q_k^m}(q_k^l \boldsymbol{C}_{k|k}^l + q_k^m \boldsymbol{C}_{k|k}^m) + \frac{q_k^l q_k^m}{(q_k^l + q_k^m)^2}(\boldsymbol{x}_{k|k}^l - \boldsymbol{x}_{k|k}^m)(\boldsymbol{x}_{k|k}^l - \boldsymbol{x}_{k|k}^m)^{\mathrm{T}} .$$

After collapsing both components are marked as used and the algorithm proceeds with the largest unused component. The procedure is repeated until the number of components does not exceed the maximal number M. After an initial phase, the number of components oscillates between M and $2M$.

In the general case, a more sophisticated clustering algorithm is required. We propose to use either maximin clustering [5] or Potts glass clustering [6]. Both algorithms can be implemented in a straightforward manner in less than a hundred lines of code. Note, however, that in general the full distance matrix of all components has to be computed. When the clusters have been found, the elements of a cluster are collapsed to a single Gaussian with the same mean and covariance matrix.

It is difficult to specify in advance how many components are needed for a satisfactory approximation. Therefore, some simulation work is required to optimize the maximum number of components in the posterior.

After finishing the collapsing procedure the prediction step to the next measurement surface has to be performed. This is done by applying the system equation to each component of the posterior density,

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_k) = \sum_{l=1}^{n_k} q_k^l \varphi(\boldsymbol{x}_{k+1}; \boldsymbol{x}_{k+1|k}^l, \boldsymbol{C}_{k+1|k}^l) ,$$

with

$$\boldsymbol{x}_{k+1|k}^l = \boldsymbol{F}_{k+1}\boldsymbol{x}_{k|k}^l, \quad \boldsymbol{C}_{k+1|k} = \boldsymbol{F}_{k+1}\boldsymbol{C}_{k|k}^l\boldsymbol{F}_{k+1}{}^{\mathrm{T}} .$$

If there are material surfaces between measurement surfaces k and $k+1$, the prediction has to be carried out in several steps. Each intermediate step terminates at a material surface, and the covariance matrix of multiple scattering is added to the covariance matrix of each component. If the process noise (multiple scattering and/or energy loss) in the material is also modeled by a Gaussian mixture, the updating of the current p.d.f. is slightly more complicated (see Section 5).

3. Performance of the parallel filter

In order to evaluate the possible gain in efficiency by using the parallel filter we have continued the simulation study reported in [1]. In this study we have assumed a two-component mixture model of the observation error. The track detector used in this study is rotationally symmetric w.r.t. the z-axis and consists of 12 cylindrical measurement surfaces at radii $R = 30, 35, \ldots, 80, 85$ cm. In every surface two coordinates are measured, $R\Phi$ and z, where Φ is the azimuth of the crossing point. The standard deviation of the measurement is assumed to be 0.2 mm for $R\Phi$ and 0.5 mm for z, which are typical values in actual detectors at high energy colliders. The correlation between the measurements is set to zero. The magnetic field is assumed to be homogeneous and parallel to z, resulting in a helical track model. We have used a standard sample of 10000 tracks with radii between 300 and 3000 cm, corresponding roughly to a p_T between 1 and 10 GeV at a field of 1.1 Tesla.

We have evaluated the efficiency of the parallel filter relative to the optimal linear filter systematically for a wide range of Gaussian mixture distributions of the observation error. The total variance of the observation error was the same in every case, corresponding to the standard deviations quoted above. The performance of the estimator is measured by the generalized variance of the five estimated track parameters $\boldsymbol{x}_{n|n} = (R\Phi, z, \vartheta, \varphi, 1/r)$, i.e. the determinant of the sample covariance matrix $\boldsymbol{C}$ of $\boldsymbol{x}_{n|n}$,

$$\boldsymbol{C} = \mathrm{E}[(\boldsymbol{x}_{n|n} - \boldsymbol{x}_{n,\mathrm{true}})(\boldsymbol{x}_{n|n} - \boldsymbol{x}_{n,\mathrm{true}})^{\mathrm{T}}],$$

where the expectation operator denotes the sample average. In particular, we look at the relative efficiency η which is defined as the generalized variance of the optimal linear estimate divided by the generalized variance of the parallel filter estimate.

As the detector is homogeneous, the Gaussian mixture distribution can be specified by two global parameters p and ρ, where $p \leq 1/2$ is the probability of an outlier and $\rho \geq 1$ is the ratio of standard deviations σ_1/σ_0. If the variance of the observation error is denoted by σ^2, then

$$\sigma_0^2 = \sigma^2/(1 - p + p\rho^2), \quad \sigma_1^2 = \sigma^2\rho^2/(1 - p + p\rho^2).$$

For the sake of simplicity, the same values of p and ρ are chosen for both $R\Phi$- and z-measurements.

Fig. 1 shows the relative efficiency η of the parallel filter as a function of the maximal number M of components in the posterior, for various values of p and ρ. The open circles correspond to Algorithm C as described above, whereas the open squares correspond to Algorithm D. The robust filter described in [1] is a special case of Algorithm C, corresponding to the case $M = 1$.

First we observe that Algorithm C is consistently better than Algorithm D. For larger values of p and ρ the relative efficiency of Algorithm C is nearly twice the one of algorithm D. In some cases the relative efficiency of Algorithm D even drops below 1, indicating that it is worse than the optimal linear filter. For ρ=2, the improvement above $M = 2$ is marginal; for larger values of ρ it is worthwhile to go beyond $M = 2$. For $p \geq 0.2$ and $\rho \geq 3$ the efficiency increases about linearly with M up to a value of $M = 8$.

The computational load of the parallel filter is summarized in Fig. 2. It shows the execution time of the parallel filter divided by the execution time of the optimal linear filter, again as a function of M. The additional cost of computing the Kullback–Leibler distances of the components in Algorithm C is reflected in the fact that Algorithm C (circles) is consistently slower than Algorithm D (squares). The robust filter of Ref. [1] is represented by the circle at $M = 1$. The execution time is about proportional to the maximal number of components, as is to be expected.

In order to asses the quality of the covariance matrix of the final estimate, Fig. 3 shows the normalized differences of the estimated and the true values of the final track parameters, for the case $p = 0.2$, $\rho = 3$, $M = 4$ (Algorithm C). In addition, a Gaussian has been fitted to the frequency distribution. The fit is excellent, and the fitted mean values and standard deviations are compatible with the theoretical values of 0 and 1, respectively. This proves that the covariance matrix produced by the parallel filter reproduces very well the actual second moments of the estimates. Fig. 3 also shows the probability transform of the "total chi-square". The frequency distribution is not flat but hill-shaped; the mean value is close to 0.5.

The results of this section have been obtained by using the correct model of the observation distribution in the parallel filter. It has been shown in [1] that p and ρ can be estimated from the data to a sufficient degree of accuracy.

4. The parallel smoother

In the linear Gaussian model, smoothing can be implemented in two different ways:

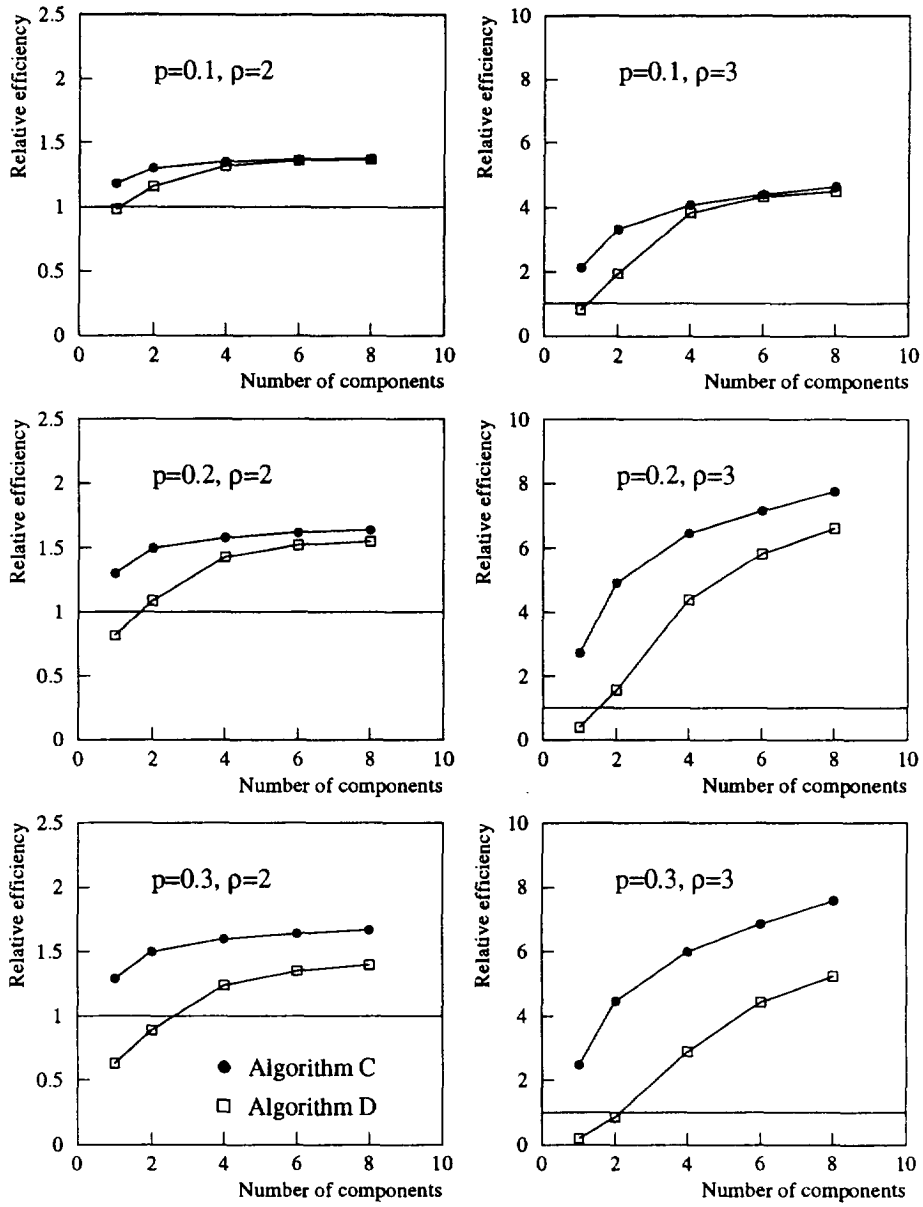

Fig. 1. The relative efficiency of the parallel filter.

- After finishing the filter, a second recursive procedure is run backwards, propagating back the full information contained in the final estimate. The algorithm can be traced back to Rauch, Tung and Striebel [7].
- Alternatively, a second filter can be run in the direction opposite to the first one. The first filter is called the forward filter, the second one the backward filter. By taking a weighted mean of forward and backward filter estimates one obtains estimates containing the information contained in all observations, i.e. smoothed estimates. Since no observation must be used more than once, one has to combine a predicted estimate of one filter with a filtered estimate of the other filter. In track fitting applications, this way of smoothing has been pioneered by P. Billoir.

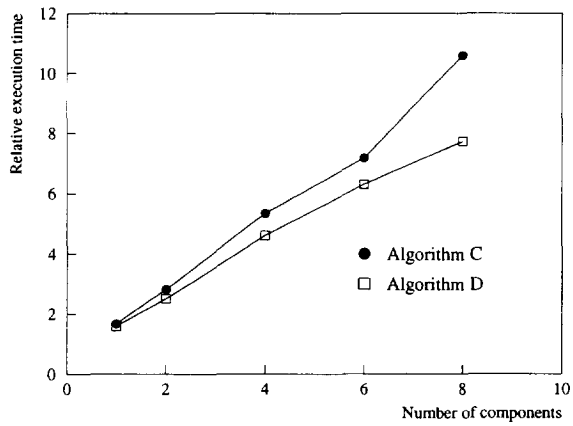

Fig. 2. The computational load of the parallel filter.

With the parallel filter, smoothing is most easily implemented by the second method. The simplest approach is to take a weighted mean of forward filter and backward filter estimates,

$$\boldsymbol{x}_{k|n} = (\boldsymbol{C}_{k|n})^{-1}[(\boldsymbol{C}_{k|k-1})^{-1}\boldsymbol{x}_{k|k-1} + (\boldsymbol{C}_{k|k,\dots,n})^{-1}\boldsymbol{x}_{k|k,\dots,n}] ,$$
$$(\boldsymbol{C}_{k|n})^{-1} = (\boldsymbol{C}_{k|k-1})^{-1} + (\boldsymbol{C}_{k|k,\dots,n})^{-1} .$$

We call this the weighted mean (WM) smoother. In the WM smoother the knowledge of the actual densities of the estimates is restricted to the first and second moment.

The problem of computing the density of the smoothed estimate in a way similar to the filter was first tackled by Kitagawa [8]. Unfortunately there is an error in his calculation. We present here the correct smoothing density. We recall that the set of observations $\{\boldsymbol{y}_1,\dots,\boldsymbol{y}_k\}$ is denoted by $\boldsymbol{Y}_k$. Now let $\boldsymbol{Y}^k$ denote the set of observations $\{\boldsymbol{y}_k,\dots,\boldsymbol{y}_n\}$. Then the prior of the forward filter at step k is given by

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) = \sum_{j=1}^{N_{k-1}} \pi_k^j \varphi(\boldsymbol{x}_k; \boldsymbol{x}_{k|k-1}^j, \boldsymbol{C}_{k|k-1}^j) .$$

The posterior of the backward filter at step k has a similar form,

$$p(\boldsymbol{x}_k|\boldsymbol{Y}^k) = \sum_{l=1}^{N_k'} \beta_k^l \varphi(\boldsymbol{x}_k; \boldsymbol{x}_{k|k,\dots,n}^l, \boldsymbol{C}_{k|k,\dots,n}^l) .$$

The posterior estimate of the backward filter can be interpreted as a virtual observation of the state $\boldsymbol{x}_k$, the p.d.f. of the observation error being given by the posterior density. Application of Bayes' theorem immediately yields the smoothing density,

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_n) \propto \sum_{j=1}^{N_{k-1}} \sum_{l=1}^{N_k'} \pi_k^j \beta_k^l \varphi(\boldsymbol{x}_k; \boldsymbol{x}_{k|k-1}^j, \boldsymbol{C}_{k|k-1}^j) \varphi(\boldsymbol{x}_k; \boldsymbol{x}_{k|k,\dots,n}^l, \boldsymbol{C}_{k|k,\dots,n}^l) .$$

Using Lemma 2 again, the smoothing density can be written in the following form:

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_n) = \sum_{j=1}^{N_{k-1}} \sum_{l=1}^{N_k'} \gamma_k^{jl} \varphi(\boldsymbol{x}_k; \boldsymbol{x}_{k|n}^{jl}, \boldsymbol{C}_{k|n}^{jl}) ,$$

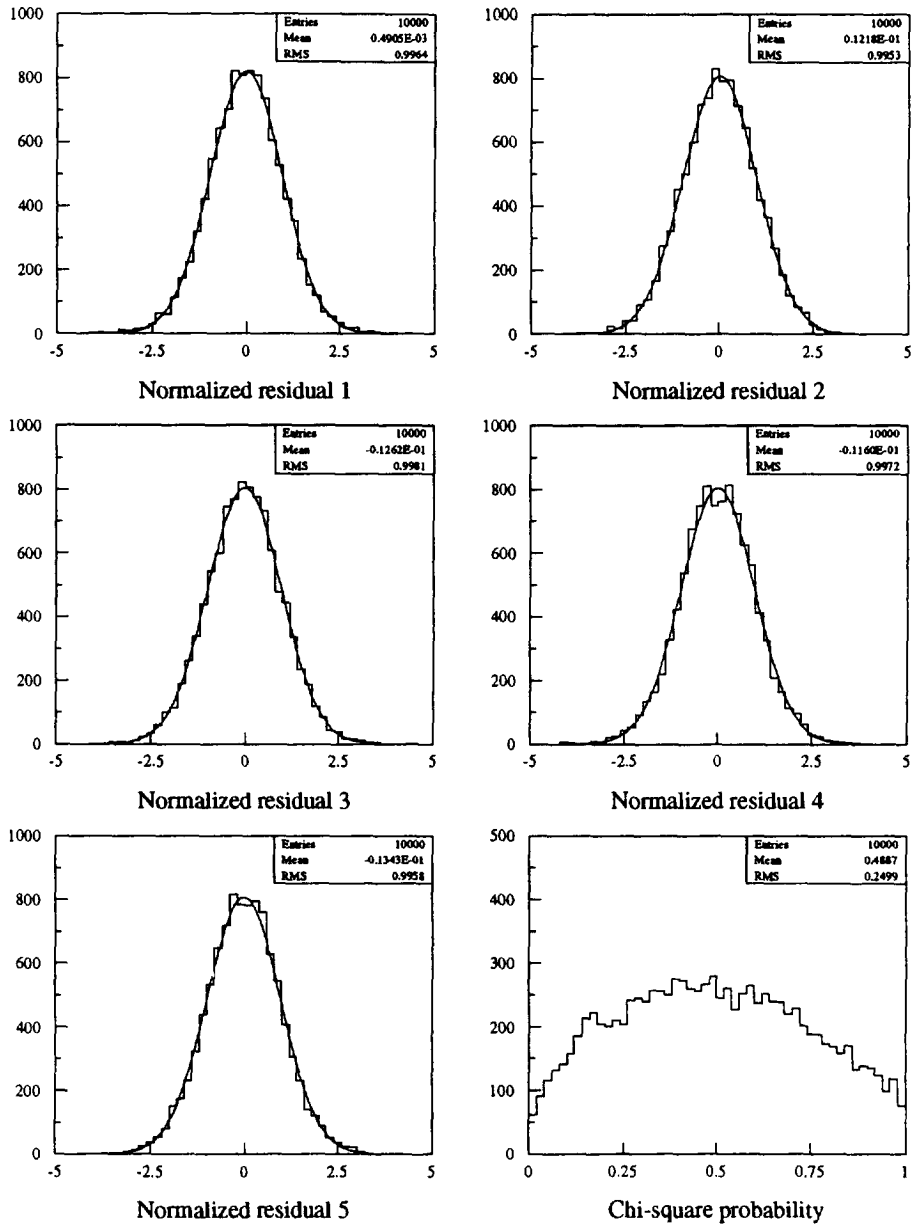

Fig. 3. Normalized residuals and chi-square probability.

the weights γ_k^{jl} being given by

$$\gamma_k^{jl} \propto \pi_k^j \beta_k^l \varphi(\boldsymbol{x}_{k|k-1}^j; \boldsymbol{x}_{k|k,\ldots,n}^l, \boldsymbol{C}_{k|k-1}^j + \boldsymbol{C}_{k|k,\ldots,n}^l) \,.$$

The constant of proportionality is determined by the requirement that the sum of all γ_k^{jl} is equal to 1. The mean $\boldsymbol{x}_{k|n}^{jl}$ and the covariance matrix $\boldsymbol{C}_{k|n}^{jl}$ of each component are the result of a Kalman filter, written here as a weighted mean,

$$x^{jl}_{k|n} = (C^{jl}_{k|n})^{-1}[(C^{j}_{k|k-1})^{-1}x^{j}_{k|k-1} + (C^{l}_{k|k,\ldots,n})^{-1}x^{l}_{k|k,\ldots,n}],$$
$$(C^{jl}_{k|n})^{-1} = (C^{j}_{k|k-1})^{-1} + (C^{l}_{k|k,\ldots,n})^{-1}.$$

The estimate $x_{k|n}$ of the Bayesian smoother and its covariance matrix $C_{k|n}$ are of course obtained as the first and second moment of the smoothing density. In [8] the mixture weights were erroneously given as

$$\gamma^{jl}_{k} = \pi^{j}_{k}\beta^{l}_{k}.$$

We call a smoother with these weights the KI smoother.

In order to compare the performance of these smoothing algorithms, we have extended the simulation study described in the preceding section. The efficiency of the nonlinear smoothers (WM, KI and BS) relative to the linear smoother is shown in Fig. 4, for various values of p and ρ. These results have been obtained with a maximum of $M = 6$ components in the posterior. We observe that BS is consistently better than WM and that its relative efficiency is much more uniform if considered as a function of the smoother step k. The KI smoother is inferior even to the WM smoother. In fact, one can ascertain by a simple example that it is biased.

Fig. 5 shows the computational load of filter and smoother as a function of the maximum number M of components in the posterior. The scale is such that the execution time of the linear filter is equal to 1. The linear smoother is a weighted mean of two linear filters and consequently takes a little more than twice as long. The Bayesian smoother (BS) takes slightly longer than the weighted mean smoother, but is clearly more efficient. Both take about twice as long as the parallel filter. This implies that the time spent in combining the two filters is only a small fraction of the time spent in the filters themselves.

5. Non-Gaussian process noise

In track fitting, process noise arises from the interaction of the charged particle with material in the detector. There are two physical processes which are the main contributors: multiple Coulomb scattering and energy loss. For a review of the underlying physics, the reader is referred to a textbook [9].

Multiple scattering results in a random deflection of the particle and, if the material traversed is sufficiently thick, also in a random offset. Let us assume that a charged particle with mass m and momentum p traverses a perfectly homogeneous layer of matter. The relevant properties of the material can be summarized in a single constant, the radiation length x_0. The actual length x of material traversed depends on the angle of incidence. Let us denote the number of radiation lengths traversed by the particle by $d = x/x_0$. As the variance of the offset is proportional to d^3, the offset is negligible for thin layers, i.e. $d \ll 1$.

We now consider the projected scattering angles θ_1, θ_2 in two directions orthogonal to each other and to the tangent of the trajectory. Because of symmetry, θ_1 and θ_2 are uncorrelated and their distribution is identical. The central limit theorem allows us to approximate this distribution by a Gaussian distribution with zero mean, again because of symmetry, and with the variance as given by the Rossi–Greisen formula,

$$\mathrm{var}(\theta_i) = kd(m^2 + p^2)/p^4, \quad k = (0.014\mathrm{GeV})^2.$$

As usual the units are chosen such that the vacuum speed of light c equals 1. Note that in some literature there is a further logarithmic correction to the variance which violates the additivity of variance for independent stochastic variables. We therefore neglect it here. Note that in the state vector the track direction is usually represented either by direction tangents $(dx/dz, dy/dz)$, or by direction cosines $(dx/ds, dy/ds)$, or by polar angle and azimuth (θ, ϕ). The covariance matrix of (θ_1, θ_2) has to be transformed accordingly.

In practice it is rarely the case that a layer of material is perfectly homogeneous. Most detectors – gaseous detectors, solid state detectors, calorimeters – have an internal cell structure, with more material concentrated in some places than in others. The same is true for most of the support structure. In the track fit it is difficult to

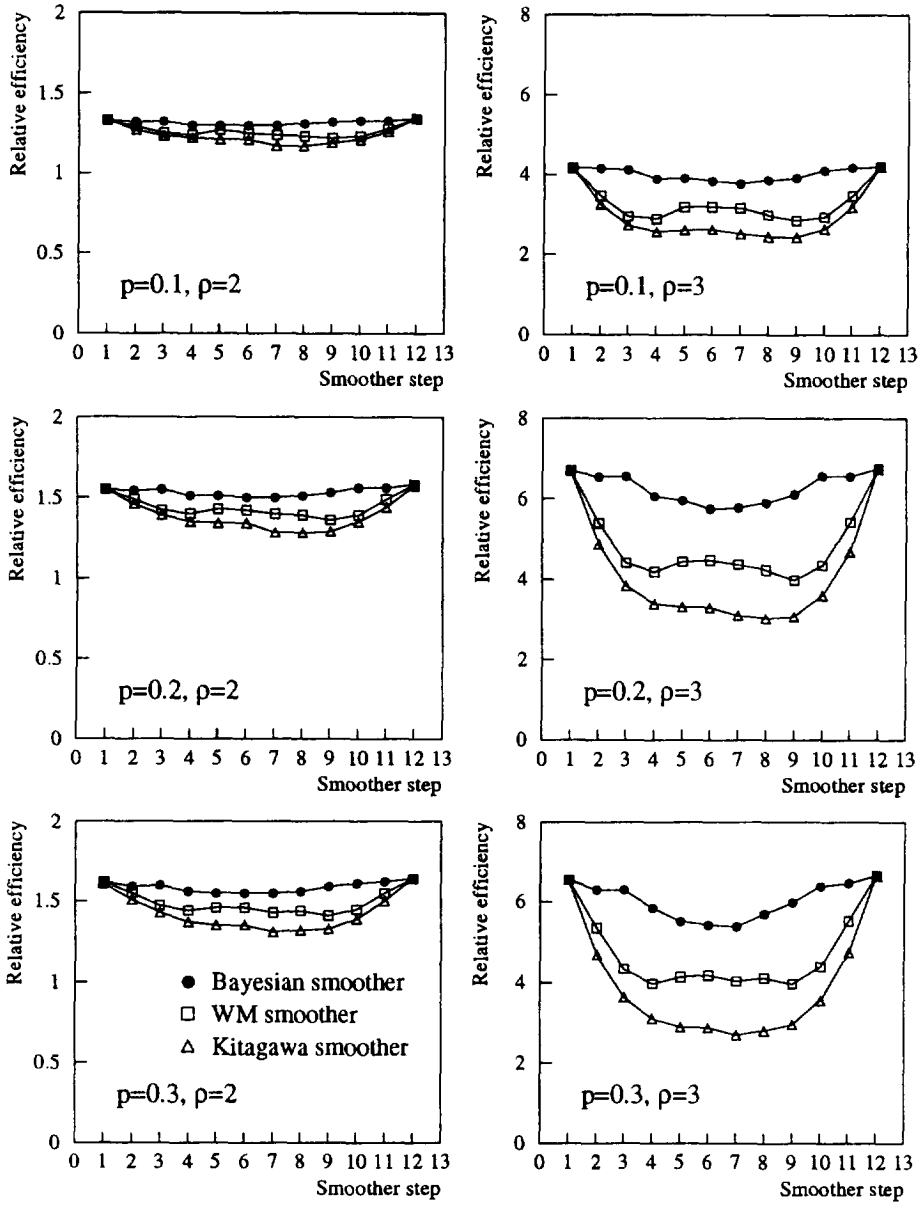

Fig. 4. The relative efficiency of the smoothing algorithms.

take this into account for two reasons. The first one is the computational load of storing and tracking through a complicated geometry. The second one is that the exact position of the trajectory is not known until after the track fit; thus it is not clear how many radiation lengths are actually traversed by the particle. A solution of this dilemma is offered by modeling multiple scattering by a Gaussian mixture with two or more components. Let us assume that multiple scattering in a thin layer of material can be modeled by the following Gaussian mixture:

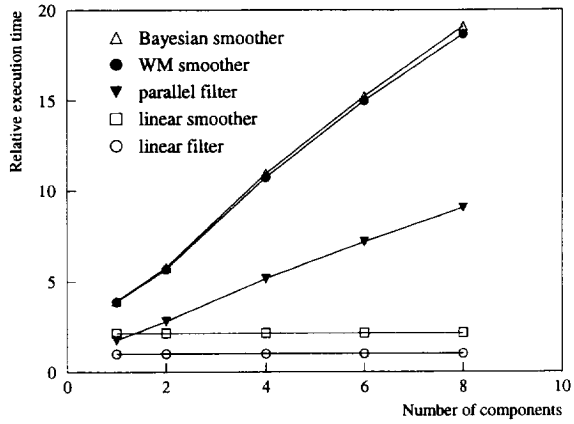

Fig. 5. The computational load of the smoother.

$$p(t) = \sum_{i=1}^{m} \alpha_i \varphi(t; \mathbf{0}, \boldsymbol{Q}_i) , \quad \sum_{i=1}^{m} \alpha_i = 1 ,$$

where $t = (t_1, t_2)$ is the vector of the two direction parameters, according to the convention of the track fit. If the p.d.f. of the state vector upon entering the layer is a Gaussian mixture of the form

$$p(\boldsymbol{x}) = \sum_{j=1}^{n} \pi_j \varphi(\boldsymbol{x}; \boldsymbol{x}_j, \boldsymbol{C}_j) ,$$

the p.d.f. of the state vector on leaving the material is given by

$$p(\boldsymbol{x}) = \sum_{j=1}^{n} \sum_{i=1}^{m} \alpha_i \pi_j \varphi(\boldsymbol{x}; \boldsymbol{x}_j, \boldsymbol{C}_j + \boldsymbol{H}^{\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{H}) ,$$

with

$$\boldsymbol{H} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} ,$$

assuming that (t_1, t_2) occupy position 3 and 4 in the state vector. If several layers of material are traversed between two measurement surfaces, reduction of the number of the components of the mixture may be required.

We now turn to the treatment of energy loss as a process noise. Minimum ionizing particles lose energy mainly by ionization of the medium. Although the energy loss is stochastic, its variance is small in this case. It is therefore the usual practice to approximate it by a constant, or to neglect it altogether.

For electrons the picture is different. Because of their small mass the dominant energy loss mechanism is bremsstrahlung. In this case the fluctuation is of the same order of magnitude as the average loss and has to be taken into account. With the linear Kalman filter the average loss and its variance can be used in the system equation [10]. This procedure would be optimal if the distribution of the energy loss were Gaussian. The p.d.f. of the electron energy E after d radiation lengths was given by Bethe and Heitler [11] as

$$p(E) = \frac{1}{E_0} \frac{[\ln(E_0/E)]^{d/\ln 2 - 1}}{\Gamma(d/\ln 2)} ,$$

where E_0 is the initial energy. This implies that $-\ln(E/E_0)$ is gamma-distributed. If we write $\Delta = (E_0 - E)/E_0$ for the (negative) relative energy loss, it is easy to see that the p.d.f. of Δ is given by

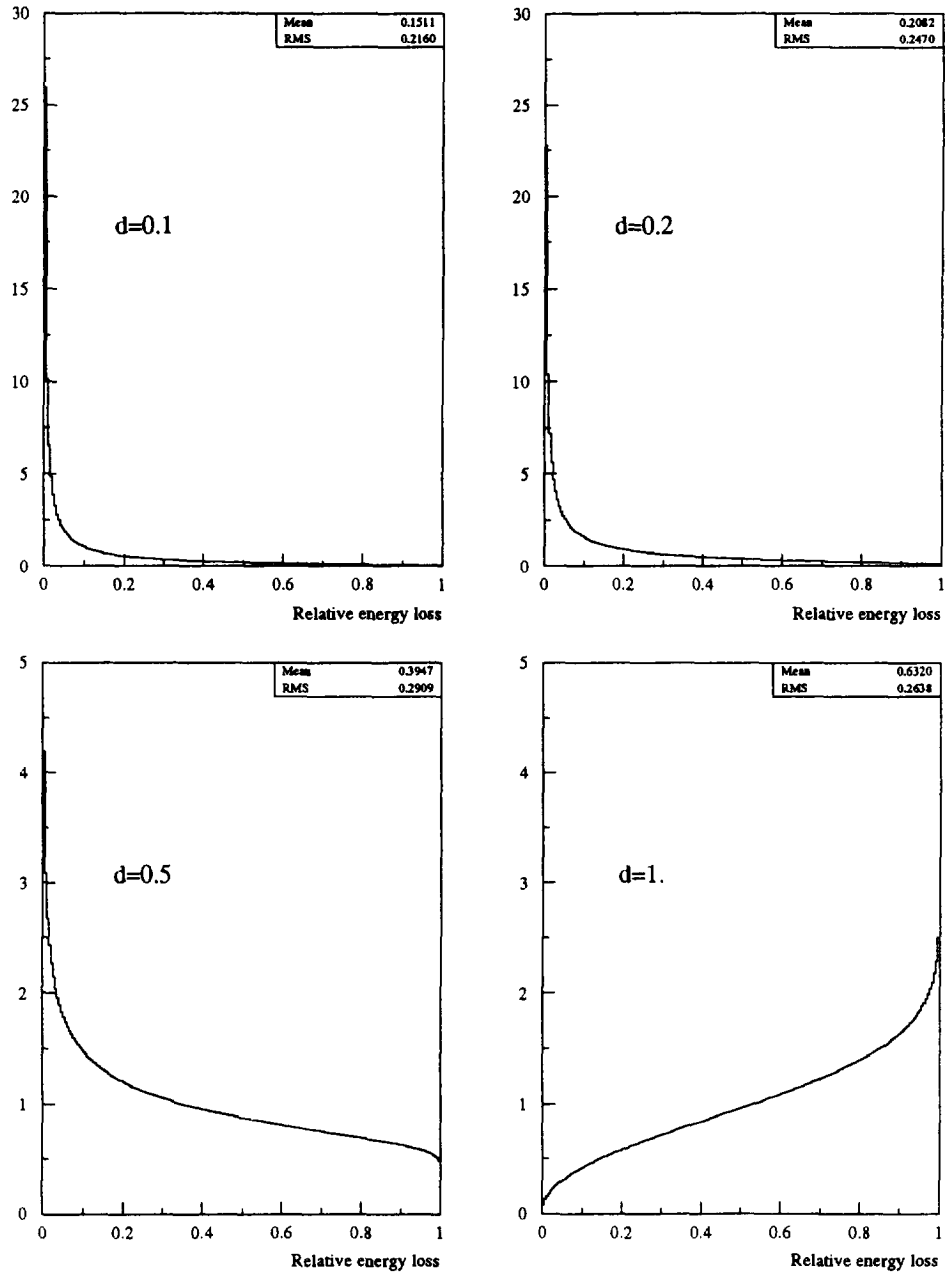

Fig. 6. Frequency distribution of the relative energy loss for electrons.

$$p(\Delta) = \frac{[-\ln(1-\Delta)]^{d/\ln 2-1}}{\Gamma(d/\ln 2)}.$$

Fig. 6 shows this p.d.f. for various values of the number d of radiation lengths.

Now it is obvious that a Gaussian is but a very poor approximation to these highly asymmetric and partly long-tailed distributions. If they can be adequately modeled by Gaussian mixtures, the reconstruction of electrons

can very likely profit from using the parallel filter. This will be the subject of a subsequent study.

6. Implementation issues

If the average number of components in the parallel filter is large, it is worthwhile to give some thought to the implementation of the basic Kalman filter algorithm. The first important point is the linear approximation of the track model, the exact track model being in most cases nonlinear. If a good approximation of the actual trajectory is available, for instance as a result of the track finding stage, it can be used as a reference track. This idea has been put forward by P. Billoir. The reference track should be an exact solution of the equation of motion. The system matrices $\boldsymbol{F}_k$ and the covariance matrices of multiple scattering are computed in the intersection points of the reference track with the respective measurement and material surfaces. The filter and the smoother operate on the differences of the actual states and the reference states. The reference track is particularly important in the initial steps of the filter, where the state may be ill defined because of an insufficient number of observations.

The second point concerns the actual computation of the prediction, filter and smoother equations. Originally, the Kalman filter was formulated in terms of covariance matrices. It can, however, be formulated as well in terms of inverse covariance or weight matrices. This is called the information filter. In order to simplify the notation, we now discard the indices which denote the current step. In the Kalman filter, the prediction step can be split into two parts: error propagation and adding process noise. A generic prediction step therefore looks like this:

$$\boldsymbol{C}' = \boldsymbol{F}\boldsymbol{C}\boldsymbol{F}^{\mathrm{T}}, \quad \boldsymbol{C}'' = \boldsymbol{C}' + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{H},$$

with $\boldsymbol{H}$ being defined as in the preceding section. In the information filter we work with the inverse covariance matrices $\boldsymbol{G} = \boldsymbol{C}^{-1}$,

$$\boldsymbol{G}' = (\boldsymbol{F}^{-1})^{\mathrm{T}}\boldsymbol{G}\boldsymbol{F}^{-1}, \quad \boldsymbol{G}'' = \boldsymbol{G}' - \boldsymbol{G}'\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{Q} + \boldsymbol{H}\boldsymbol{G}'\boldsymbol{H}^{\mathrm{T}})^{-1}\boldsymbol{H}\boldsymbol{G}'.$$

We need to invert the system matrix $\boldsymbol{F}$ for the weight propagation. Note that $\boldsymbol{F}^{-1}$ is precisely the system matrix of the backward filter. Thus if smoothing is required, $\boldsymbol{F}^{-1}$ is needed anyway. In adding the process noise, only a 2×2 matrix has to be inverted.

The filter step of the information filter is a weighted mean of the prediction and of the observation. No matrix inversion is required. It is sufficient to solve a system of linear equations in order to compute the filtered estimate. The linear information filter is therefore faster than the standard Kalman filter. In the parallel filter, however, the covariance matrices of the components have to be computed whenever the collapsing procedure is activated. In this case it is therefore better to work throughout with the covariance matrices.

Appendix A

Lemma 1. Let $\boldsymbol{A}$ be matrix of dimension $m \times n$ and $\boldsymbol{B}$ a matrix of dimension $n \times m$. Then

$$|\boldsymbol{I}_m + \boldsymbol{A}\boldsymbol{B}| = |\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{A}| .$$

Proof. Without loss of generality we may assume that $m \leq n$. If $\boldsymbol{AB}$ has the eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$ then $\boldsymbol{BA}$ has the eigenvalues $\{\lambda_1, \ldots, \lambda_m, 0, \ldots, 0\}$ (see, e.g., [12, page 163]). It is easy to show that $\boldsymbol{I}_m + \boldsymbol{AB}$ has the eigenvalues $\{1 + \lambda_1, \ldots, 1 + \lambda_m\}$, and that $\boldsymbol{I}_n + \boldsymbol{BA}$ has the eigenvalues $\{1 + \lambda_1, \ldots, 1 + \lambda_m, 1, \ldots, 1\}$. As the determinant of a square matrix is the product of its eigenvalues, the identity is established. □

Lemma 2. Let $\varphi(\boldsymbol{y}; \boldsymbol{Hx}, \boldsymbol{V})$ be the Gaussian density function of the observation error and let $\varphi(\boldsymbol{x}; \boldsymbol{x}_P, \boldsymbol{C}_P)$ be the prior Gaussian density function of the predicted state vector. Then

$$\varphi(\boldsymbol{y}; \boldsymbol{Hx}, \boldsymbol{V})\varphi(\boldsymbol{x}; \boldsymbol{x}_P, \boldsymbol{C}_P) = \varphi(\boldsymbol{y}; \boldsymbol{Hx}_P, \boldsymbol{V} + \boldsymbol{HC}_P\boldsymbol{H}^T)\varphi(\boldsymbol{x}; \boldsymbol{x}_F, \boldsymbol{C}_F) ,$$

with

$$\boldsymbol{x}_F = \boldsymbol{x}_P + \boldsymbol{C}_F\boldsymbol{H}^T\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{Hx}_P) = \boldsymbol{C}_F(\boldsymbol{C}_P^{-1}\boldsymbol{x}_P + \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{y}) ,$$
$$\boldsymbol{C}_F^{-1} = \boldsymbol{C}_P^{-1} + \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{H} .$$

Thus $\boldsymbol{x}_F$ is the filtered estimate (weighted mean) and $\boldsymbol{C}_F$ is its covariance matrix.

Proof. We first show that the exponents on both sides are identical. Now the exponent on the left-hand side equals

$$(\boldsymbol{y} - \boldsymbol{Hx})^T\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{Hx}) + (\boldsymbol{x} - \boldsymbol{x}_P)^T\boldsymbol{C}_P^{-1}(\boldsymbol{x} - \boldsymbol{x}_P) ,$$

and the exponent on the right-hand side equals

$$(\boldsymbol{y} - \boldsymbol{Hx}_P)^T(\boldsymbol{V} + \boldsymbol{HC}_P\boldsymbol{H}^T)^{-1}(\boldsymbol{y} - \boldsymbol{Hx}_P) + (\boldsymbol{x} - \boldsymbol{x}_F)^T\boldsymbol{C}_F^{-1}(\boldsymbol{x} - \boldsymbol{x}_F) .$$

Using the definition of $\boldsymbol{x}_F$ and $\boldsymbol{C}_F$ and the identity

$$(\boldsymbol{V} + \boldsymbol{HC}_P\boldsymbol{H}^T)^{-1} = \boldsymbol{V}^{-1} - \boldsymbol{V}^{-1}\boldsymbol{H}(\boldsymbol{C}_P^{-1} + \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{V}^{-1}$$

and by some reshuffling of terms the identity of the exponents is established. It follows that

$$\varphi(\boldsymbol{y}; \boldsymbol{Hx}, \boldsymbol{V})\varphi(\boldsymbol{x}; \boldsymbol{x}_P, \boldsymbol{C}_P) = k \cdot \varphi(\boldsymbol{y}; \boldsymbol{Hx}_P, \boldsymbol{V} + \boldsymbol{HC}_P\boldsymbol{H}^T)\varphi(\boldsymbol{x}; \boldsymbol{x}_F, \boldsymbol{C}_F) ,$$

with

$$k = \frac{|\boldsymbol{V} + \boldsymbol{HC}_P\boldsymbol{H}^T|^{1/2}|\boldsymbol{C}_F|^{1/2}}{|\boldsymbol{V}|^{1/2}|\boldsymbol{C}_P|^{1/2}} = \frac{|\boldsymbol{I} + \boldsymbol{HC}_P\boldsymbol{H}^T\boldsymbol{V}^{-1}|^{1/2}}{|\boldsymbol{I} + \boldsymbol{H}^T\boldsymbol{V}^{-1}\boldsymbol{HC}_P|^{1/2}} .$$

From Lemma 1 it follows that $k = 1$. □

Acknowledgements

I want to thank W. Mitaroff for many useful comments on the manuscript.

References

[1] R. Frühwirth, Comput. Phys. Commun. 85 (1995) 189.
[2] D. Peña and I. Guttman, Commun. Statist. Theory Meth. 18 (1989) 817.
[3] G. Kitagawa, Comput. Math. Appl. 18 (1989) 503.
[4] R. Frühwirth, Nucl. Instr. Meth. A 262 (1987) 444.
[5] E.A. Patrick, Fundamentals of Pattern Recognition (Prentice-Hall, Englewood Cliffs, NJ, 1972).
[6] M. Bengtsson and P. Roivainen, Int. J. Neural Systems 6 (1995) 119.
[7] H.E. Rauch, F. Tung and C.T.Striebel, AIAA J. 3 (1965) 1445.

[8] G. Kitagawa, Ann. Inst. Statist. Math. 46 (1994) 605.
[9] R.K. Bock et al., Data Analysis Techniques for High-Energy Physics Experiments (Cambridge Univ. Press, Cambridge 1990).
[10] D. Stampfer, R. Frühwirth and M. Regler, Comput. Phys. Commun. 79 (1994) 157.
[11] H.A. Bethe and W. Heitler, Proc. R. Soc. London A 146 (1934) 83.
[12] R. Zurmühl and S. Falk, Matrizen und ihre Anwendungen, Teil 1: Grundlagen, 5th ed. (Springer, New York, Heidelberg, Berlin, 1984).