A Kalman-Particle Kernel Filter and its Application to Terrain Navigation

Dinh-Tuan Pham
Laboratoryof Modeling and Computation, CNRS
BP 53X, 38041 Grenoble cedex 9, France.
dinh-tuan.pham@imag.fr

Karim Dahia
The French National Aerospace Research Establishment (ONERA)
Chemin de la Hunière, 91761 Palaiseau cedex, France
karim.dahia@onera.fr

Christian Musso
The French National Aerospace Research Establishment (ONERA)
BP72 - 29 avenue de la Division Leclerc 92322 Châtillon cedex, France
christian.musso@onera.fr

Abstract – *A new nonlinear filter, the Kalman- Particle Kernel Filter (KPKF) is proposed. Compared with other particle filters like Regularized Particle Filter (RPF), it adds a local linearization in a kernel representation of the conditional density. Therefore, it strongly reduces the number of redistributions which causes undesirable Monte Carlo fluctuations. This new filter is applied to terrain navigation, which is a nonlinear and multimodal problem. Simulations show that the KPKF outperforms the classical particle filter.*

Keywords: Kalman filter, kernel density estimator, regularized particle filter, Inertial navigation System (INS), Cramer Rao bound, Monte Carlo method, bandwidth, terrain navigation.

1 Introduction

The Kalman filter provides a computationally efficient (recursive) solution to the least square estimator of the states of the dynamical system. However, if the system and/ or measurement equations are nonlinear (which is often the case), an extended Kalman filter (EKF) needs to be used, in which the above equations are linearized around the current system state. In this case the filter is no longer optimal (in the mean squares sense) and more importantly it can diverge if the nonlinearity is too strong. One can of course try to implement the *optimal nonlinear filter* but it is impractical because it require multiple integration in a high dimensional space (which needs to be numerically approximated anyway). Therefore a so called *particle filter* has been proposed [1,2], which can be viewed as discrete stochastic approximation to the optimal filter. However, this filter can be very costly to implement, as a very large number of particles is usually needed, especially in high dimensional system. In case of low dynamical noise, we observe that in multiplying the high weighted particles, the prediction step will explore poorly the state space. The particle clouds will concentrate on few points of the state space. This phenomenon is called *particle degeneracy*, and causes the divergence of the filter.

Recently, Musso et Al. [3] have introduced the Regularized particle filter (RPF), which is based on regularization of the empirical distribution associated with particle system, using the kernel method, see [4]. The main idea consists in changing the discrete approximation to a continuous approximation in the particle filter such that the resampling step is changed into simulations from an absolutely continuous distribution, hence producing a new particle system with different particle locations. Nevertheless, these techniques have sometimes weaknesses in spaces of high dimension : compromises have to be found between the increase in the number of the particles (which decreases the error of Monte-Carlo drawings) and the need to limit this number (to have a reasonable computing time). In this paper, we introduce a new filter called Kalman-particle kernel filter, which can combine the benefice of the correction in the Kalman filter (both in terms of computational efficiency and filter performance) and the robustness of the regularized particle filter with regard to nonlinearities of the system and/or measurement equations.

In section II, we provide a brief presentation of the nonlinear filtering problem and the theoretical optimal filter. We then describe briefly the particle and regularized particle implementation in next section. Our Kalman-particle filter kernel filter is introduced and discussed in section IV. Finally, the last section provides an example of application to the terrain navigation problem with elevation measurements.

2 The nonlinear filtering problem

Consider a stochastic dynamical system in which the state vectors sequence $\{X_k\}$ is described by

$$X_k = F_k(X_{k-1}) + W_{k-1} \qquad (1)$$

where F_k is a possibly nonlinear function from R^n to R^m, n and m being the dimension of the state and the

observation vectors respectively, and $\{W_k\}$ is an iid (independent identically distributed) process with covariance matrix S_k. The goal of the filtering problem is to estimate (recursively) the state X_k from the measurements :

$$Y_k = H_k(X_k) + V_k \tag{2}$$

up to time k, where H_k is a possibly nonlinear function from $\boldsymbol{R^n}$ to $\boldsymbol{R^m}$ and $\{V_k\}$ is an iid process with covariance matrix R_k, m being the dimension of the observation vector. By adopting the mean square error criterion, the best estimate of the state vector in known to be its conditional expectation given $Y_1,...,Y_{k-1}$. However, point estimate may be not enough, one would need to know the accuracy of the estimate and thus the conditional covariance matrix of X_k given $Y_1,...,Y_k$ is also of interest. But in the nonlinear case, the conditional distribution of X_k given $Y_1,...,Y_k$ is not Gaussian (in general) and hence its means and covariance matrix alone may not provide enough useful information. Therefore, it is of interest to consider this conditional distribution itself. Besides, one would need to work with it anyway in order to construct *a recursive* filtering algorithm. We denote by $P_k(dx_k \mid y_1,...,y_k)$ the conditional distribution of X_k given the observations $y_1,...,y_k$ of $Y_1,...,Y_k$ and by $P_{k|k-1}(dx_k \mid y_1,...,y_{k-1})$ the conditional (predictive) distribution of X_k given the observations $y_1,...,y_k$. The optimal nonlinear filtering algorithm consists of two steps.

- The prediction step :

At time $k-1$, it is supposed that the conditional distribution $P_{k-1}(dx_{k-1} \mid y_1,...,y_{k-1})$ is already available. The prediction step consists in computing the predictive distribution $P_{k|k-1}(dx_k \mid y_1,...,y_{k-1})$ via the Chapman Kolmogorov equation

$$P_{k|k-1}(dx_k \mid y_1,...,y_{k-1}) = \int_{R^n} G\left[dx_k \mid F_k(u), S_k(u)\right] P_{k-1}(du \mid y_1,...,y_{k-1}) \tag{3}$$

where $G(dx \mid m, \sum)$ denotes the Gaussian distribution with mean $\boldsymbol{m}$ and covariance matrix $\sum$. In practice, the matrix S_k is often nonsingular. By (4), the predictive distribution then admits a Gaussian density :

$$P_{k|k-1}(x_k \mid y_1,...,y_{k-1}) = \int_{R^n} \phi\left[x_k - F_k(u) \mid S_k(u)\right] P_{k-1}(du \mid y_1,...,y_{k-1}) \tag{4}$$

where

$$\phi(x \mid \sum) = \exp[-(x^T \sum^{-1} x)/2] / \sqrt{\det(2\pi \sum)}$$

denote the density of the Gaussian distribution of zero mean and covariance matrix $\sum$.

- The correction step :

This step computes via Bayes's rule the new conditional distribution $P_k(dx_k \mid y_1,...,y_k)$, once a new observation y_k is available. To this end, one first computes the joint predictive (conditional) distribution of X_k, Y_k, given the observations $y_1,...,y_{k-1}$, which is given by

$$P_k(dx_k dy_k \mid y_1,...,y_k) = P_{k|k-1}(dx_k \mid y_1,...,y_{k-1}) G(dy_k \mid H_k(x_k), R_k)$$

Then one would decompose this joint predictive distribution as a "product" of a marginal distribution of Y_k (conditionally on the observations $y_1,...,y_{k-1}$), denoted by $P^Y_{k|k-1}(dy_k \mid y_1,...,y_{k-1})$, and the conditional distribution of X_k given Y_k (also conditionally on the observations $y_1,...,y_{k-1}$), which is no other than $P_k(dx_k \mid y_1,...,y_k)$,

$$P_k(dx_k dy_k \mid y_1,...,y_k) = P^Y_{k-1}(dy_k \mid y_1,...,y_{k-1}) P_k(dx_k \mid y_1,...,y_k)$$

In the case where both S_k and R_k are invertible, the above formula takes the form much easier to understand. Indeed, the joint predictive distribution $P_k(dx_k / y_{1,...,}y_k)$ then admits the density

$$p_k(x_k, y_k \mid y_1,...,y_{k-1}) = p_{k|k-1}(x_k \mid y_1,...,y_{k-1})\, \phi(y_k - H_k(x_k) \mid R_k) \tag{5}$$

The conditional distribution of interest $P_k(dx_k \mid y_1,...,y_k)$ then also admits a density which is simply the ratio of the above joint to the marginal density,

$$p(x_k \mid y_1,...,y_k) = \frac{p_{k|k-1}(x_k \mid y_1,...,y_{k-1})\,\phi[y_k - H_k(x_k) \mid R_k]}{\int p_{k|k-1}(x_k \mid y_1,...,y_{k-1})\,\phi[y_k - H_k(x_k) \mid R_k]\,dx} \quad (6)$$

3 The particle filter and the regularized particle filter

The particle filter can be viewed as an approximation to the above optimal filter by using the Monte-Carlo method to approximate the conditional or predictive distribution by a mixture of Dirac distribution. We describe here the simplest form of this filter. Assume that at the time k-1, one has approximation to the conditional distribution $P_{k-1}(dx_{k-1} \mid y_1,...,y_{k-1})$ of X_k given $\{y_1,...,y_{k-1}\}$ of the form $\sum_{i=1}^{N} \omega_{k-1}^i \delta(dx_{k-1} \mid x_{k-1}^i)$ where ω_{k-1}^i are positive weights summing to 1, $x_{k-1}^1,...,x_{k-1}^N$ are points in R^n (called particles) and $\delta(dx_{k-1} \mid x_{k-1}^i)$ denotes the Dirac distribution with mass at x_{k-1}^i. The predictive distribution admits a density which is simply a mixture of Gaussian density (4)

$$p_{k|k-1}(x_k \mid y_1,...,y_{k-1}) = \sum_{i=1}^{N} \omega_{k-1}^i \phi[x_k - F_k(x_{k-1}^i) \mid S_k^i]$$

One again approximates this mixture by a mixture of Dirac distribution $\sum_{i=1}^{N} \omega_{k|k-1}^i \delta(. \mid x_{k|k-1}^i)$ with the same weights $\omega_{k|k-1}^i = \omega_{k-1}^i$ and located at $x_{k|k-1}^i$, where $x_{k|k-1}^i$ are obtained from $F_k(x_{k-1}^i)$ by adding independent Gaussian vectors of mean zero and covariance matrices S_k^i. Then from the previous subsection, the conditional distribution of X_k given $Y_1 = y_1,...,Y_k = y_k$, is the weighted distribution,

$$\omega_k^i = \frac{\omega_{k|k-1}^i \phi[y_k - H_k(x_{k|k-1}^i) \mid R_k(x_{k|k-1}^i)]}{\sum_{j=1}^{N} \omega_{k|k-1}^j \phi[y_k - H_k(x_{k|k-1}^j) \mid R_k(x_{k|k-1}^j)]}$$

and at the points $x_k^i = x_{k|k-1}^i$. The Sequential Importance Sampling (SIS) [1] algorithm consists of recursive propagations of the weights and support points at each new measurement.

3.1 Degeneracy Problem

A common problem with the SIS particle filter is the degeneracy phenomenon, where after a few iterations, all but a few particles will have negligible weight. A good measure of degeneracy is the entropy (7) of the distribution of the particle weights. Resampling is performed if the entropy is too low, more precisely if

$$Ent = \log N + \sum_{i=1}^{N} \omega_{k|k-1}^i \log \omega_{k|k-1}^i \quad (7)$$

exceeds a threshold [5].

3.2 The Regularized particle filter

The Regularized particle filter is identical to the particle filter except for the resampling stage [3]. The RPF resamples from a continuous approximation of the posterior density

$$\hat{p}(x_k \mid y_1,...,y_k) = \sum_{i=1}^{N} \omega_k^i K_h(x_k - x_k^i) \quad (8)$$

where

$$K_h(x) = \frac{1}{h^n} K(\frac{x}{h}) \quad (9)$$

in the re-scaled Kernel density $K(.)$, $h > 0$ is the kernel bandwidth (a scalar parameter), n is the dimension of the state vector x, and $\omega_k^i, i = 1,...,N$ are particles weights. The kernel K and bandwidth h are chosen to minimize the Mean Integrated Square Error (MISE) between the true posterior density and the corresponding regularized empirical representation in (7), defined as

$$MISE(\hat{p}) = E\{\int [\hat{p}(x_k \mid y_{1:k}) - p(x_k \mid y_{1:k})]^2 dx_k\}$$

where $y_{1:k}$ is a shorthand for $y_1,...,y_k$ and $\hat{p}(x_k \mid y_{1:k})$ denotes the approximation to $p(x_k \mid y_{1:k})$ given by (7). In the special case of the all the samples having the same weight, the optimal choice of the kernel is the Epanechnikov kernel [3]. For the sake of homogeneity with our new filter, we use the Gaussian kernel, wich is almost optimal.

$$K(x) = \frac{1}{(2\pi)^{n/2}} \exp(-\frac{1}{2}\|x\|^2) \quad (10)$$

The corresponding optimal bandwidth is

$$h_{opt} = AN^{-1/(n+4)} \quad (11)$$

$$A = \mu\,[4/(n+2)]^{(1/(n+4))} \qquad (12)$$

where μ is a tuning parameter related to the multimodality of the conditional density.

4 The Kalman-Particle Kernel Filter

This filter combines the good properties of the Kalman filter and the regularized particle filter. The main idea is to represent the (predictive) probability density as a mixture of Gaussian densities of the form

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} \phi(x - x_i \mid P^i) \qquad (13)$$

where $\{x_1,...,x_N\}$ are a set of particles, P^i are positive definite matrices. In classical kernel density estimation, one takes $P^i = P$ equal to h^2 times the sample matrix of the particle $x_1,...,x_N$, h being a parameter to be adjusted. But the structure (13) with covariance matrices P^i being of the order h^2 may not be preserved in time. In order that it is so, we introduce 2 kinds of resampling : the partial and full resamplings. Partial resampling is performed to limit the Monte Carlo fluctuations, in the case where the particle weights are nearly uniform so as there is little risk of degeneracy.

The filter algorithm consists of 4 steps.

- The initialization step :

We initialize the algorithm at the first correction (and not prediction), that is we initialize the density $\hat{p}_{1|0}$. Based on the kernel density estimation method, we simply draw N particles $x^1_{1/0},...,x^N_{1/0}$ from the unconditional distribution of X_1 and take $\hat{p}_{1|0}(x_{1|0}) = (1/N)\sum_{i=1}^{N}\phi(x^i_{1|0} \mid P_{1|0})$, where $P_{1|0}$ equals h^2 times the sample covariance matrix of the particles.

- The correction step :

According to formula (5), the joint predictive density of X_k, given $\{Y_1 = y_1,...,Y_{k-1} = y_{k-1}\}$, is

$$p_k(x_k, y_k \mid y_1,...,y_{k-1}) =$$

$$\sum_{i=1}^{N} \omega^i_{k|k-1}\phi(x_k - x^i_{k|k-1} \mid P^i_{k|k-1})\phi(y_k - H_k(x_k) \mid R_k)$$

This is mixture of distribution of densities

$$\phi(x_k - x^i_{k|k-1} \mid P^i_{k|k-1})\phi(y_k - H_k(x_k) \mid R_k) \qquad (14)$$

Since $P^i_{k|k-1}$ is small, $\phi(x_k - x^i_{k|k-1} \mid P^i_{k|k-1})$ becomes negligible as soon as x_k is not close to $x^i_{k|k-1}$. Thus in (14) one can linearize H_k around $x^i_{k|k-1}$. Which yields that (14) can be approximated by

$$\phi(x_k - x^i_{k|k-1} \mid P^i_{k|k-1})\,\phi(y_k - y^i_{k|k-1} + \nabla H^i_k(x_k - x^i_{k|k-1}) \mid R^i_k) \qquad (15)$$

where $y^i_{k|k-1} = H_k(x^i_{k|k-1})$ and ∇H^i_k denotes the gradient (matrix) of H_k at the point $x^i_{k|k-1}$. It can be shown, using similar calculations as in the derivation of the Kalman filter, that (15) can be re-factorized as

$$\phi(x_k - x^i_k \mid P^i_k)\phi(y_i - y^i_{k|k-1} \mid \Sigma^i_k)$$

where

$$x^i_k = x^i_{k|k-1} + G^i_k(y_k - y^i_{k|k-1}) \qquad (16)$$

$$G^i_k = P^i_{k|k-1}\nabla H_k^{i\,T}(\Sigma^i_k)^{-1} \qquad (17)$$

$$P^i_k = P^i_{k|k-1} - P^i_{k|k-1}\nabla H_k^{i\,T}(\Sigma^i_k)^{-1}\nabla H^i_k P^i_{k|k-1} \qquad (18)$$

$$\Sigma^i_k = \nabla H^i_k P^i_{k|k-1}\nabla H_k^{i\,T} + R_k \qquad (19)$$

Therefore

$$p_k(x_k, y_k \mid y_1,...,y_{k-1}) \approx$$

$$\sum_{i=1}^{N} \omega^i_{k|k-1}\phi(x_k - x^i_k \mid P^i_k)\,\phi(y_k - y^i_{k|k-1} \mid \Sigma^i_k)$$

The conditional density $p_k(x_k \mid y_1,...,y_k)$ of X_k given the observations $Y_1 = y_1,...,Y_k = y_k$, being proportional to $p_k(x_k, y_k \mid y_1,...,y_{k-1})$, is thus given by

$$p_k(x_k \mid y_1,...,y_k) = \sum_{i=1}^{N} \omega^i_k \phi(x_k - x^i_k \mid P^i_k) \qquad (20)$$

where

$$\omega_k^i = \frac{\omega_{k|k-1}^i \phi(y_k - y_{k|\backslash k-1}^i \mid \Sigma_k^i)}{\sum_{j=1}^N \omega_{k|k-1}^j \phi(y_k - y_{k|k-1}^j \mid \Sigma_k^j)} \quad (21)$$

One see that the conditional density $p_k(x_k \mid y_1,...,y_k)$ is also a mixture of Gaussian densities, as states before. Note that the covariance matrices P_k^i of the components of this mixture, by (18), is bounded above by $P_{k|k-1}^i$, hence remain small they are so before. Finally, one can interpret this correction step as composed of two types of correction : a Kalman type correction defined by (17), (19) and a particle type correction defined by (19) and (21).

- The prediction step :

The correction step has provided an approximation to the conditional density $p_k(x_k \mid y_1,...,y_k)$ in the form of a mixture of Gaussian density (20) with the mixture component matrices $P_{k|k-1}^i$ being small. By (4) and (20), the predictive density at the next step equals

$$p_{k+1|k}(x_{k+1} \mid y_1,...,y_k) =$$
$$\sum_{i=1}^N \omega_k^i \int_{R^n} \phi(x_{k+1} - F_{k+1}(u) \mid S_{k+1})\, \phi(u - x_k^i \mid P_k^i)$$

but since $\phi(u - x_k^i \mid P_k^i)$ becomes negligible a soon as u is not close to x_k^i, one can again make the approximation $F_{k+1}(u) \approx F_{k+1}(x_k^i) + \nabla F_{k+1}^i (u - x_k^i)$ where ∇F_{k+1}^i denotes the gradient (matrix) of F_{k+1} at the point x_k^i. Using this approximation, it can be shown that,

$$p_{k+1|k}(x_{k+1} \mid y_1,...,y_k) =$$
$$\sum_{i=1}^N \omega_k^i\, \phi(x_k - F_{k+1}(x_k^i) \mid \nabla F_{k+1}^i P_k^i \nabla F_{k+1}^{i\,T} + S_{k+1})$$

Thus the predictive density is still a mixture of Gaussian distribution, with the covariance matrix of the i-th component of the mixture equal to

$$P_{k+1|k}^i = \nabla F_{k+1}^i P_k^i \nabla F_{k+1}^{iT} + S_{k+1} \quad (22)$$

and with the weights $\omega_{k+1|k}^i = \omega_k^i$. However, the mixture component covariance matrices may not be small. This may be due to presence of the additive term S_{k+1} and the amplification effect of the multiplication by F_{k+1}^i.

- The resampling step :

To reduce the errors introduced by resampling, we adopt a simple rule, which waits for *m* filter cycles before possibly resampling (*m* being a tuning parameter). In these *m* filter cycles resampling is skipped. One simply set $x_{k+1|k}^i = F_{k+1}(x_k^i)$ and $\omega_{k+1|k} = \omega_k^i$. After these cycles a full or partial resampling will be made depending on the entropy criterion (7). It purpose is both to keep the matrices $P_{k+1|k}^i$ low and to avoid degeneracy. To perform resampling, one first computes the matrices :

$$\Pi_{k+1|k} = \sum_{i=1}^N \omega_k^i P_{k+1|k}^i + \mathrm{cov}(F_{k+1}(x_k) \mid \omega_k)$$

where $\mathrm{cov}(F_{k+1}(x_k) \mid \omega_k)$ denotes the sample covariance matrix of the vectors $F_{k+1}(x_k^i)$ relative to the weights ω_k^i, $i = 1,...,N$. Then one computes its Cholesky factorization $\Pi_{k+1|k} = CC^T$ and then h^{*2}, the minimum of the smallest eigenvalues of the matrices $C^{-1} P_{k+1|k}^i (C^T)^{-1}$. The next calculations depend on whether partial or full resampling is required.

<u>*Partial resampling*</u> *:* if the weights are nearly uniform, that is, if the entropy criterion is less than the threshold (7), we do :

For each i add to $F_{k+1}(x_k^i)$ a random Gaussian vector with zero mean and covariance matrix $P_{k+1|k}^i - h^* \Pi_{k+1|k}$ to obtain the new particles $x_{k+1|k}^i$. Then set $\omega_{k+1|k}^i = \omega_k^i$.

<u>*Full resampling*</u> *:* if the weights are disparate, that is the entropy criterion is greater than the threshold (7), we do :

Select N particles among $F_{k+1}(x_k^1),\ldots,F_{k+1}(x_k^N)$ according to the probabilities $\omega_k^1,...,\omega_k^N$, then add to each of them a random Gaussian vector with zero mean

and covariance matrix $P^i_{k+1|k} - h^* \Pi_{k+1|k}$ to get the new particles $x^i_{k+1|k}$. Then set $\omega^i_{k+1|k} = 1/N$.

After resampling, the matrices $P^i_{k+1|k}$ are reset to $h^2 \Pi_{k+1|k}$. Thus they become small and one may expect that they remain so for least several subsequent filter cycles.

5 Application to terrain navigation with elevation measurements

5.1 Problem description

Aircraft autonomous navigation can be made with Inertial navigation System (INS). Absolute positions and velocities are obtained by integrating the gyroscopics/accelerometrics measurements. But the position error grows with time. Therefore external measurements are needed to correct these errors. A radioaltimeter provides elevation measurements (relatives heights Y_k) along the aircraft path. Comparing these elevations with a Digital Terrain Elevation Data (DTED) on board it is theorically possible to reconstruct the absolute position of the aircraft . The DTED gives the absolute elevation $H(x_k, y_k)$ as function of the latitude/longitude coordinates x_k and y_k of the aircraft. This yields the measurement equation :

$$Y_k = z_k - h(x_k, y_k) + V_k = H(X_k) + V_k \qquad (23)$$

where z_k is the altitude of the aircraft and V_k a centered Gaussian noise.

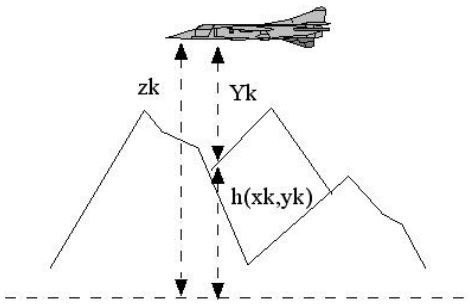

Figure 1 : Elevation measurements

This method is called terrain navigation ([6], [8]) and is useful if the terrain contains enough relevant information, that is if the elevation variation is noticeable. The above equation should be supplemented by the dynamical equations which describes the movement of the aircraft in term of its acceleration provided by the accelerometers. These equations are rather complex as one has to take into account the curvature and the rotation of the earth and the variation of the "angles" of the aircraft provided by the gyroscopic measurements. The gyroscopic and accelerometric error then appear indirectly as dynamical errors. In the simulation below, we however consider a simple model in which the aircraft has zero acceleration up to some Gaussian error. The dynamical equations can then be written (in discrete time) as

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + \Delta t \begin{bmatrix} vx_k \\ vy_k \\ vz_k \end{bmatrix}, \quad \begin{bmatrix} vx_k \\ vy_k \\ vz_k \end{bmatrix} = \begin{bmatrix} vx_{k-1} \\ vy_{k-1} \\ vz_{k-1} \end{bmatrix} + W_k \quad (24)$$

where vx_k, vy_k, vz_k denote the velocities of the aircraft, Δt the time interval between measurements and W_k represents the acceleration error. Thus X_k, Y_k in equations (1) and (2) corresponds to $X_k = [x_k\, y_k\, z_k\, vx_k\, vy_k\, vz_k]$ (24) and Y_k (23). The function F_k is linear and does not depend on k and $H_k = H$. Our simplified model still captures the essence of the terrain navigation problem, since the difficulty in such problem lies in the divergence of the dynamic, the strong nonlinearity of the measurement equation and the multimodal terrain.

5.2 Simulation results

We present an application of the KPKF in terrain navigation [6]. Simulations have been performed on a DTED (see Figure 8) with angle 15 arc second resolution. The region of interest for the simulation is centered on a point at coordinates (44 deg N, 3 deg. E). The 6 dimensional state $X_k = [x_k\, y_k\, z_k\, vx_k\, vy_k\, vz_k]$ is to be estimated. This problem contains strong multimodality due to the ambiguity of the terrain. The initial position of the aircraft is a Gaussian centered on the reference position (200 km, 150 km) (see Figure 8). The initial state uncertainty (which is large) is given by the covariance matrix,

$P_0 = diag(3000\,m, 3000\,m, 100\,m, 5\,m/s, 5\,m/s, 1\,m/s)^2$

The reference trajectory has been generated according to the model (24) with

$\mathrm{cov}(W_k) = diag(\Delta t\sigma_x, \Delta t\sigma_y, \Delta t\sigma_z)^2$, with an initial horizontal velocity vector of magnitude 141 m/s $(vx_0 = 100m/s, vy_0 = 100m/s)$ and with vertical velocity vector equals to 4 m/s. The number of measurements is 400 (see Figure 2), the number of particles for RPF is 10 000, for KPKF is 1300 which gives the same computing time for the 2 filters. Every 0.7 s the aircraft measures the elevation with standard deviation fixed to 15 m, according to equation (23). 100 Monte Carlo (MC) trials have been performed. In averaging the results of the filters, we compute the RMS

(Root Mean Square) for each filter. The PCRB (Posterior Cramer-Rao bound) has been computed. It is an universal lower bound for the covariance matrix for any unbiased estimators [6].

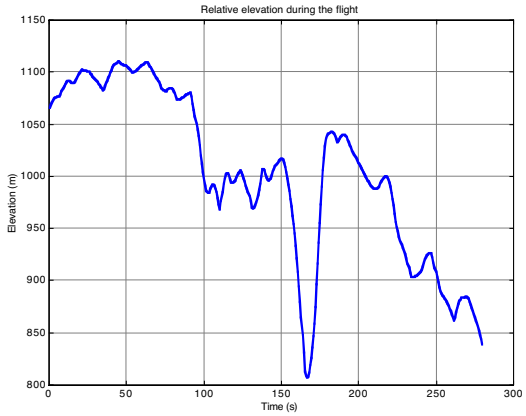

Figure 2 : Relative elevation during the flight.

The tuning parameters of the 2 filters are :

RPF : Best value of threshold is *0.3* (7*)*. The bandwidth constant is $\mu = 0.25$ (12)

KPKF : Best value of threshold is 0.1. The bandwidth is $\mu = 1$ and tuning parameter $m = 35$ (filter cycle, before resampling).

Results for 2 filters are shown on following figures. For each trial, and for each measurement time, the aircraft position is estimated by the mean of the particle cloud. The RPF has given 5 divergences (out of 100), the KPKF gives 2 divergences. We call divergence when the state estimate at the three last steps of the algorithm is out the 99% confidence region (ellipsoid) given by the PCRB. The rough terrain case (see Figure 6, 7) allows good performances of the 2 filters (especially for the KPKF) with a convergence rate of the RMS close to the PCRB.

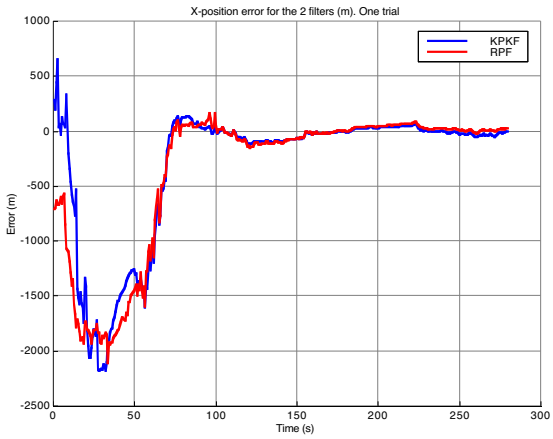

Figure 3. One trial x-error for the 2 filters

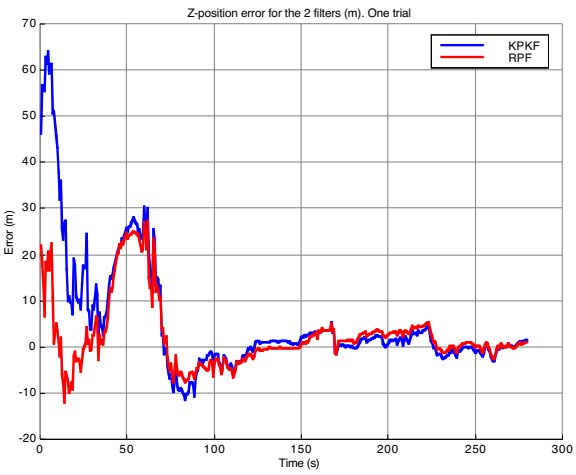

Figure 4. One trial z-error for the 2 filters

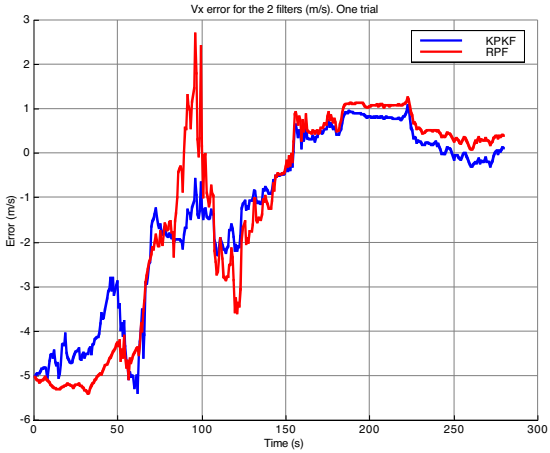

Figure 5. One trial vx-error for the 2 filters

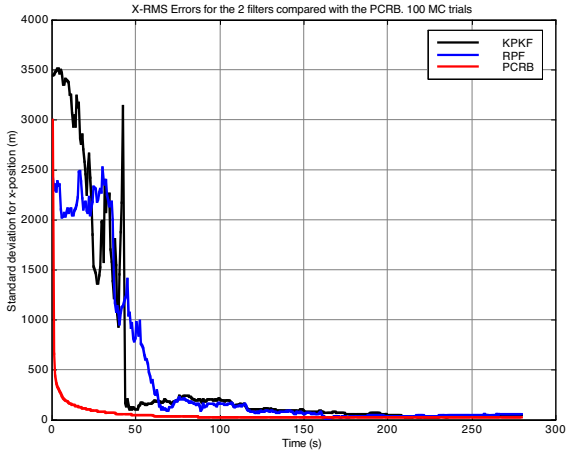

Figure 6. RMS, PCRB -x error

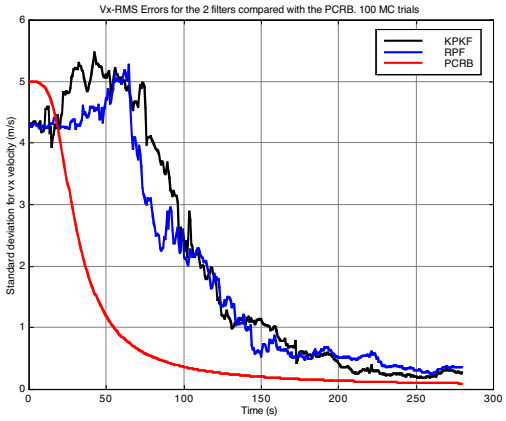

Figure 7. RMS, PCRB -vx error

6 Conclusions

We have proposed a new particle filter called Kalman-Particle Kernel Filter (KPKF). It is based on kernel representation of the conditional density and on local linearization. It reduces the number of redistributions which causes undesirable Monte Carlo fluctuations. Simulations in a terrain navigation context show the robustness of this filter. In this difficult context with a large initial position uncertainty, the particle filter KPKF works with only 1300 particles

References

[1] D. Salmond, N .Gordon, « Group and extended object tracking » SPIE Conference on Signal and Data Processing of small targets, vol 3809, pp. 282-296, Colorado, July 1999.

[2] A. Doucet, S.J. Godsill, C. Andrieu, « On sequential Simulation based methods for Bayesian filtering ». Statistics and Computing, vol. 10, no. 3, pp. 197-208, 2000.

[3] C. Musso, N. Oudjane, F. LeGland, « Improving Regularized particle filters in : Sequential Monte Carlo Methods in Practice ». ch. 12, pp. 247-271. Springer Verlag, New York, 2001.

[4] B.W. Silverman. « Density estimation for statistics and data analysis ». Edition Chapman & Hall, 1986.

[5] D. T. Pham, « Stochastic methods for sequential data assimilation in strongly nonlinear systems », Monthly Weather Review, vol.129, no.5, pp. 1194-1207, 2001.

[6] N. Bergman. Ph. D. thesis. « Recursive Bayesian Estimation, Navigation and tracking Application ». Linköping University, Sweden, 1999.

[7] P. Tichavsky, C.H. Muravchik, A. Nehorai. « Posterior Cramer-Rao Bounds for discrete-time nonlinear filtering ». IEEE, Vol 46, N° 5, pp. 1386-1396, May, 1998.

[8] N. Bergman, L. Ljung, F. Gustafsson. « Terrain navigation using Bayesian statistics ». IEEE Control System Magazine, 19(3), pp. 33-40, 1999. http://www.control.isy.liu.se/publications

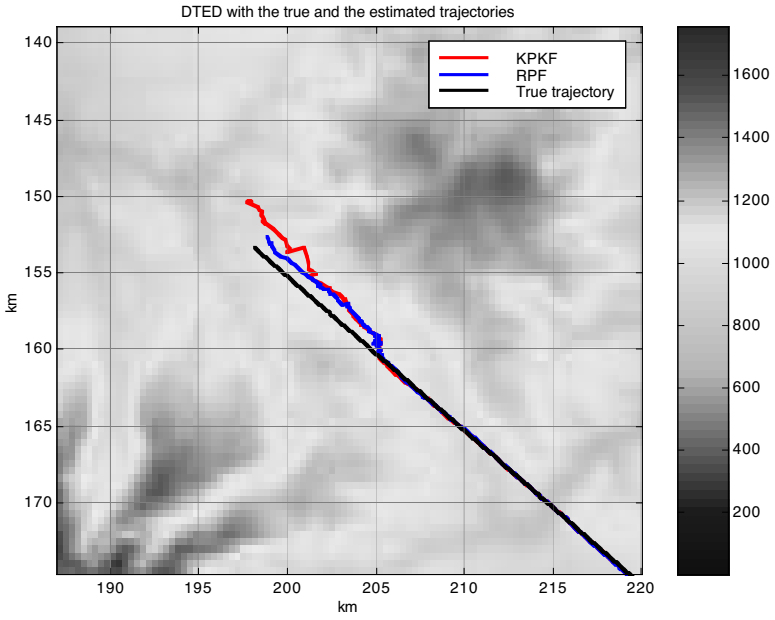

Figure 8. DTED with aircraft trajectory