APPLICATION OF THE KALMAN-PARTICLE KERNEL FILTER TO THE UPDATED INERTIAL NAVIGATION SYSTEM

Karim Dahia[†], *Christian Musso*[†], *Dinh Tuan Pham*[‡], *Member, IEEE and Jean-Pierre Guibert*[†]

[†]The French National Aerospace Research Estabishment (ONERA), Chemin de la Hunière, 91761, Palaiseau cedex, France, email: karim.dahia@onera.fr.
[‡]Laboratory of Modeling and Computation (CNRS), BP 53X, 38041 Grenoble cedex 9, France.

ABSTRACT

This paper considers a new nonlinear filter which combines the good properties of the Kalman filter and the particle filter. Compared with other particle filters like Rao-Blackwellised particle filter (RBPF), it adds a local linearization in a kernel representation of the conditional density, which yields a Kalman type correction complementing the usual particle correction. Therefore, it can operate with much less number of particles. It reduces the Monte-Carlo fluctuations and the risk of divergence. The new filter is applied to the highly nonlinear and multimodal terrain navigation problem. Simulations show that it outperforms the RBPF.

1. INTRODUCTION

The Kalman filter has been a very successful and useful tool in a wide range of engineering problems, but it has been designed under the assumption of a linear model. Although its linearization version, called extended Kalman filter (EKF), provides a sensible solution in the nonlinear case, the EKF is no longer optimal and can even diverge if too strong nonlinearities are present. The particle filter (see [4, 8, 10, 12] for a review) has been introduced to deal with such situations. This kind of filter has appeared under several names in the literature, such as interacting particle (IPF) [9], Bootstrap filter and Sampling/Importance Resampling (SIR) [5], Kernel filter [6] and Sequential Monte Carlo methods [8]. However, the filter can be very costly to implement, as a very large number of particles is usually needed, especially in high dimensional system. Further, after some filtering steps, many particle weights can become negligible and the filter would then poorly explore the state space as the particles cloud is concentrated on a few points. This phenomenon is called the particle degeneracy and often leads to filter divergence.

Recently, Musso *et al.* [10] have introduced the Regularized Particle Filter (RPF), which is based on the regularization of the empirical distribution associated with the particles cloud, using the kernel density estimation method [14]. Our paper describes and completes a new filter we have introduced in [13], called Kalman-Particle Kernel Filter (KPKF), which combines the efficiency of the extended Kalman filter (both in terms of computational cost and performance) and the robustness of the RPF. The main idea is to linearize the system and measurement equations around a set of particles and apply a local "Kalman type" correction to each particle based on these linearized equations. Such correction is complemented by the classical "particle type" correction which consists in redistributing the weights of the particles to reflect the change of their likelihood as a new observation is available. The advantage of the Kalman correction is that it tends to pull the particles toward the true system state, thereby the redistribution of weights in the "particle type" correction would be more spread out among the particles. This would reduce the risk of degeneracy and allow the filter to operate reliably with much fewer particles. A "hybrid" Kalman-particle filter known as Rao Blackwellised particle filter (RBPF) has been proposed [3]. Our KPKF is different and more general since it does not rely on a partial linearity assumptions. Another recent hybrid filter called the Gaussian Sum Particle filtering [7] also uses local Kalman filtering similar to the KPKF. But the derivation of our filter is different, it uses the kernel decomposition of the predictive density, which justifies properly the local linearization (see section 2). Furthermore, the KPKF uses an original resampling procedure which reduces the Monte-Carlo fluctuations. We present an application of our filter to the terrain navigation problem. The KPKF is compared with the popular RBPF.

2. THE KALMAN-PARTICLE KERNEL FILTER

In the sequel, we shall consider the nonlinear stochastic discrete time dynamical system:

$$\mathbf{x}_k = F_k(\mathbf{x}_{k-1}) + \mathbf{w}_k \quad (1)$$

$$\mathbf{y}_k = H_k(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

where $\mathbf{x}_k$ is the unobserved state vector (to be estimated), of dimension n, $\mathbf{y}_k$ is the observation vector, of dimension m, F_k and H_k are two continuously differentiable maps from $\mathbb{R}^n$ to $\mathbb{R}^n$ and from $\mathbb{R}^n$ to $\mathbb{R}^m$ and $\mathbf{w}_k$ and $\mathbf{v}_k$ denote the dynamical and observation Gaussian noise vectors with associated covariance matrices respectively $\mathbf{S}_k$ and $\mathbf{R}_k$. The initial state vector x_0 is assumed to be random with a known distribution. The aim is to compute the filtering density $p_k(\mathbf{x}_k|\mathbf{y}_{1:k})$, which is the conditional density of the state vector at each time k given the measurements $\mathbf{y}_{1:k} = (\mathbf{y}_1, ..., \mathbf{y}_k)$ up to time k. For this purpose, the predictive density $p_{k|k-1}(\mathbf{x}_k|\mathbf{y}_{1:k-1})$, which is the conditional density of the state vector at time k given the measurement before time k, will be needed. The main idea of the KPKF is to approximate this density by a mixture of Gaussian densities with *small* covariance matrix as in the kernel density method [14] (with a small bandwidth):

$$p_{k|k-1}(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \sum_{i=1}^{N} \omega^i_{k|k-1}\phi(\mathbf{x}_{k|k-1} - \mathbf{x}^i_{k|k-1}|\mathbf{P}^i_{k|k-1}) \quad (3)$$

where $\mathbf{x}^i_{k|k-1}$ are particles in $\mathbb{R}^n$, $\omega^i_{k|k-1}$ are probabilities (particles weights) and $\phi(.|\mathbf{P})$ denotes the centered Gaussian density with covariance matrix $\mathbf{P}$. The matrices $\mathbf{P}^i_{k|k-1}$ are assumed to be small.

The filter is initialized by generating sample according to the law of x_0. Then the filter cycle consists of 3 steps.

2.1 The correction step

The main point is that starting with a predictive density of the form (3), the filtering density of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is also of the same form (up to some approximations). Indeed, the joint predictive density of $\mathbf{x}_k$ and $\mathbf{y}_k$ is

$$p_k(\mathbf{x}_k,\mathbf{y}_k|\mathbf{y}_{1:k-1}) = p_{k|k-1}(\mathbf{x}_k|\mathbf{y}_{1:k-1})\phi[\mathbf{y}_k - H_k(\mathbf{x}_k)|\mathbf{R}_k]$$

Hence, by (3) the above right hand side equals

$$\sum_{i=1}^N \omega^i_{k|k-1}\phi(\mathbf{x}_k - \mathbf{x}^i_{k|k-1}|\mathbf{P}^i_{k|k-1})\phi[\mathbf{y}_k - H_k(\mathbf{x}_k)|\mathbf{R}_k]$$

But since $\mathbf{P}^i_{k|k-1}$ *is small*, $\phi(\mathbf{x}_k - \mathbf{x}^i_{k|k-1}|\mathbf{P}^i_{k|k-1})$ will become negligible as soon as $\mathbf{x}_k$ is *not close* to $\mathbf{x}^i_{k|k-1}$ and thus in $\phi(\mathbf{x}_k - \mathbf{x}^i_{k|k-1}|\mathbf{P}^i_{k|k-1})\phi[\mathbf{y}_k - H_k(\mathbf{x}_k)|\mathbf{R}_k]$, one can linearize the map H_k around $\mathbf{x}^i_{k|k-1}$ and approximate this term by

$$\phi(\mathbf{x}_k - \mathbf{x}^i_{k|k-1}|\mathbf{P}^i_{k|k-1})\phi[\mathbf{y}_k - \mathbf{y}^i_{k|k-1} + \mathbf{H}^i_k(\mathbf{x}_k - \mathbf{x}^i_{k|k-1})|\mathbf{R}_k] \quad (4)$$

where $\mathbf{y}^i_{k|k-1} = H_k(\mathbf{x}^i_{k|k-1})$ and $\mathbf{H}^i_k$ denotes the gradient matrix of H_k at the point $\mathbf{x}^i_{k|k-1}$. One can then apply to (4) a similar calculation as in the derivation of the Kalman filter and thus obtain finally:

$$p_k(\mathbf{x}_k,\mathbf{y}_k|\mathbf{y}_{1:k-1}) \approx \sum_{i=1}^N \omega^i_{k|k-1}\phi(\mathbf{x}_k - \mathbf{x}^i_k|\mathbf{P}^i_k)\phi(\mathbf{y}_k - \mathbf{y}^i_{k/k-1}|\mathbf{\Sigma}^i_k)$$

where

$$\mathbf{x}^i_k = \mathbf{x}^i_{k|k-1} + \mathbf{G}^i_k(\mathbf{y}_k - \mathbf{y}^i_{k|k-1}) \quad (5)$$

$$\mathbf{P}^i_k = \mathbf{P}^i_{k|k-1} - \mathbf{P}^i_{k|k-1}\mathbf{H}^{iT}_k(\mathbf{\Sigma}^i_k)^{-1}\mathbf{H}^i_k\mathbf{P}^i_{k|k-1} \quad (6)$$

$$\mathbf{G}^i_k = \mathbf{P}^i_{k|k-1}\mathbf{H}^{iT}_k(\mathbf{\Sigma}^i_k)^{-1} \quad (7)$$

$$\mathbf{\Sigma}^i_k = \mathbf{H}^i_k\mathbf{P}^i_{k|k-1}\mathbf{H}^{iT}_k + \mathbf{R}_k \quad (8)$$

The filtering density $p_k(\mathbf{x}_k|\mathbf{y}_{1:k})$, being proportional to $p_k(\mathbf{x}_k,\mathbf{y}_k|\mathbf{y}_{1:k-1})$, is given by

$$p_k(\mathbf{x}_k|\mathbf{y}_{1:k}) = \sum_{i=1}^N \omega^i_k\phi(\mathbf{x}_k - \mathbf{x}^i_k|\mathbf{P}^i_k) \quad (9)$$

where

$$\omega^i_k = \frac{\omega^i_{k|k-1}\phi(\mathbf{y}_k - \mathbf{y}^i_{k|k-1}|\mathbf{\Sigma}^i_k)}{\sum_{j=1}^N \omega^j_{k|k-1}\phi(\mathbf{y}_k - \mathbf{y}^j_{k|k-1}|\mathbf{\Sigma}^j_k)} \quad (10)$$

One sees that the filtering density $p_k(\mathbf{x}_k|\mathbf{y}_{1:k})$ is also *a mixture of Gaussian densities*, as stated before. Note that the covariance matrices $\mathbf{P}^i_k$ of the components of this mixture (6) are bounded above by $\mathbf{P}^i_{k|k-1}$, hence *remain small* if they are so before. Finally, one can interpret the correction step as composed of two types of correction: a Kalman-type correction defined by (5) and a particle type correction defined by (10).

2.2 The prediction step

From the filtering density (9) and the dynamical equation (1), one gets the predictive density:

$$p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) =$$

$$\sum_{i=1}^N \omega^i_k \int_{\mathbb{R}^n} \phi[\mathbf{x}_{k+1} - F_{k+1}(\mathbf{u})|\mathbf{S}_{k+1}]\phi(\mathbf{u} - \mathbf{x}^i_k|\mathbf{P}^i_k)d\mathbf{u} \quad (11)$$

but since $\phi(\mathbf{u} - \mathbf{x}^i_k|\mathbf{P}^i_k)$ becomes negligible as soon as $\mathbf{u}$ is not close to $\mathbf{x}^i_k$, one can again make the approximation $F_{k+1}(\mathbf{u}) \approx F_{k+1}(\mathbf{x}^i_k) + \mathbf{F}^i_{k+1}(\mathbf{u} - \mathbf{x}^i_k)$ where $\mathbf{F}^i_{k+1}$ denotes the gradient (matrix) of F_{k+1} at the point $\mathbf{x}^i_k$. With this approximation, the integral in the above right hand side appears as the density of the sum of a Gaussian random vector of mean $F_{k+1}(\mathbf{x}^i_k)$ and covariance matrix $\mathbf{S}_{k+1}$ and an independent Gaussian vector with mean zero and covariance matrix $\mathbf{F}^i_{k+1}\mathbf{P}^i_k\mathbf{F}^{iT}_{k+1}$. Therefore

$$p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) \approx \sum_{i=1}^N \omega^i_k\phi[\mathbf{x}_{k+1} - F_{k+1}(\mathbf{x}^i_k)|\mathbf{P}^i_{k+1|k}] \quad (12)$$

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

$$\mathbf{P}^i_{k+1|k} = \mathbf{F}^i_{k+1}\mathbf{P}^i_k\mathbf{F}^{iT}_{k+1} + \mathbf{S}_{k+1} \quad (13)$$

and the weights $\omega^i_{k+1|k} = \omega^i_k$. However, the above covariance matrix is usually greater than that of previous step, either due to the presence of the additive term $\mathbf{S}_{k+1}$ and/or the amplification effect of the multiplication by $\mathbf{F}^i_{k+1}$.

2.3 The resampling step

For any particle filter the weights ω^i_k degenerate with time: eventually only a few particles have a non negligible weight. A good measure of degeneracy is the following entropy criterion (*Ent*) which is the difference between the entropy of the uniform distribution and that of the particles weights. Degeneracy occurs when *Ent* if greatest than some threshold η.

$$Ent = \log N + \sum_{i=1}^N \omega^i \log \omega^i \quad (14)$$

On the other hand, the matrices $\mathbf{P}^i_{k+1|k}$ can increase in the prediction step and thus become large eventually, which invalidates the linearization. Therefore, we shall resort to resampling to approximate (12) by

$$p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) \approx \sum_{i=1}^N \omega^i_{k+1|k}\phi[\mathbf{x}_k - \mathbf{x}^i_{k+1|k}|\mathbf{P}_{k+1|k}]$$

with a different set of particles $\mathbf{x}^i_{k+1|k}$ and weights $\omega^i_{k+1|k}$ and a small matrix $\mathbf{P}_{k+1|k}$. Classical resampling [5] (called here full resampling), in which the particle weights are redistributed uniformly, is normally performed. But when *Ent* is less η (meaning that the distribution of weights is already close to uniform), one can perform partial resampling which retains the same weights to reduce the Monte Carlo fluctuations. The rationale of our resampling procedure will be detailed in a technical report (downloadable at $www-lmc.imag.fr/lmc-sms/Dinh-Tuan.Pham/publics-triees.html$), only the procedure is described here due to space limitation. Let $\mathbf{\Pi}_{k+1|k} = \sum_{i=1}^{N} \omega_k^i \mathbf{P}_{k+1|k}^i + \mathrm{cov}[F_{k+1}(\mathbf{x}_k)|\omega_k]$ (where $\mathrm{cov}(.|\omega_k)$ is the sample covariance matrix relative to the probabilities ω_k^i), be the covariance matrix of the density (12) and h^* be the greatest value such that $\mathbf{P}_{k+1|k}^i - h^{\star 2}\mathbf{\Pi}_{k+1|k} > 0$ for all i. A value of $h > h^*$ will be computed from h^* according to a certain rule (it can also be adjusted as a tuning parameter). Then draw N random Gaussian vectors U_i ($i = 1,...,N$) with zero mean and covariance matrix $\mathbf{P}_{k+1|k}^i - h^{\star 2}\mathbf{\Pi}_{k+1|k}$.

- Full resampling: ($Ent > \eta$) select N particles among $F_{k+1}(\mathbf{x}_k^1), \dots, F_{k+1}(\mathbf{x}_k^N)$ according to the probabilities $\omega_k^1, \dots, \omega_k^N$, then add to each of them the above random vector U_i to get the new particles $\mathbf{x}_{k+1|k}^i$. Then set $\omega_{k+1|k}^i = 1/N$ and $\mathbf{P}_{k+1|k} = h^2 \Pi_{k+1|k}$.
- Partial resampling: ($Ent \leq \eta$) simply add to the $F_{k+1}(\mathbf{x}_k^i)$ the above vector U_i to obtain the new particles $\mathbf{x}_{k+1|k}^i$. Then set $\omega_{k+1|k}^i = \omega_k^i$ and $\mathbf{P}_{k+1|k} = h^2 \mathbf{\Pi}_{k+1|k}$.

In practice, we found that without resampling the matrices $\mathbf{P}_{k+1|k}^i$ and the entropy criterion increase (with a rate depending on the application) slowly with the filter cycles. Therefore we adopt a simple rule to perform full or partial resampling after $\boldsymbol{m}$ filter cycles. During these cycles only local Kalman filters are done.

3. APPLICATION TO TERRAIN NAVIGATION

The particle filters have already been applied in terrain navigation [2, 11]. This application context, where the state vector is of high dimension and the filtering density is often multimodal, can be considered as a good benchmark to compare the performance of different particle filters. Therefore, we shall apply the KPKF to this problem and compare it with the widely used RBPF.

3.1 Problem formulation

Position, velocity and attitude information are provided by Inertial Navigation Systems (INS), based on inertial properties of on-board sensors such as spring-mass accelerometers and gyrometers. The inertial rotational motion angular velocity is measured by gyrometers, the specific force measured by accelerometers. The integration of the navigation equations provides position ($R_N^{INS}, R_E^{INS}, R_D^{INS}$) North, East and Down, and velocity estimates. However, the errors of these estimates grow with time, therefore INS must be recalibrated periodically to maintain reliable navigation quality [11]. The KPKF will directly estimate the inertial error with altimeter measurements.

3.2 Dynamical Model

The dynamical model for the inertial error δx can be modeled by a set of first order linear differential equations:

$$d[\delta x(t)]/dt = F(t)\delta x(t) + w(t)$$

where $\delta x(t)$ is the 15 dimensional state vector to be estimated, $w(t)$ denotes the noise term. The state vector includes 9 navigation parameters, 3 accelerometer bias parameters and 3 gyrometer bias parameters. The following error model, called Ψ angle error model [1], is often used:

$$\begin{cases} d\Psi/dt = (\rho+\Omega)\wedge\Psi - \varepsilon_g \\ d[\delta V]/dt = -\Psi\wedge f + \varepsilon_a + \delta g - (\rho+2\Omega)\wedge\delta V \\ d[\delta R]/dt = \delta V - \rho\wedge\delta R \\ d[b_a]/dt = w_a \\ d[b_g]/dt = w_g \end{cases} \tag{15}$$

where all the variables are expressed in the navigation frame (North, East, Down), Ψ is the attitude (roll, pitch, yaw) error vector, δV is the velocity error vector, δR is the position error vector ($\delta R_N, \delta R_E, \delta R_D$), ρ is the angular velocity of rotation of the navigation frame w.r.t the Earth, Ω is the rotation velocity of the Earth, f is the specific force, ε_a and ε_g are the errors vectors of the accelerometers and gyrometers measurements (which depend linearly on b_a and b_g), δg is the gravity error vector. w_a and w_g are Gaussian noise vectors. The notations $\wedge$ denotes the vector product.

3.3 Measurement Model

A radioaltimeter provides elevation measurements (the relatives heights y_k see Fig.1) along the aircraft path. Comparing on board these elevations with a Digital Terrain Elevation Data (DTED), it is possible to reconstruct the absolute position of the aircraft, if there is enough relevant information in the elevation variation. The DTED gives the absolute elevation as function of the latitude/longitude. This gives the following measurement equation:

$$y_k = R_D^{INS} + \delta R_D - DTED(R_N^{INS} + \delta R_N, R_E^{INS} + \delta R_E) + v_k \tag{16}$$

where v_k is the altimeter sensor error.

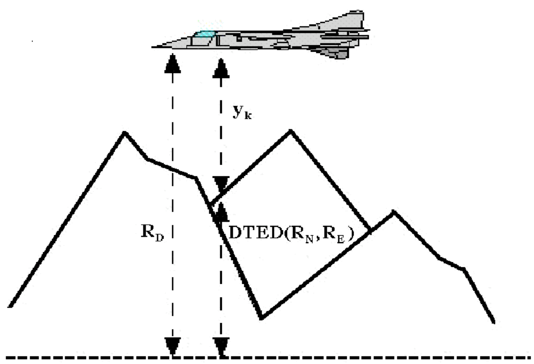

Figure 1: Elevation measurement in terrain navigation

3.4 Simulation results

The initial state uncertainty is 1000 m for R_N and R_E, 100 m for R_D, $5m/s$ for V_N and V_E, $1m/s$ for V_D, 0.5 deg for the attitude angles, $0.01m/s^2$ for the accelerometer bias and $0.06deg/s$ for the gyrometers bias. The horizontal velocity vector is $250m/s$. The number of measurements is 400, the number of particles 15000 for the RBPF is and 2500 for the

KPKF, which gives the same computing time for 2 filters. Every 0.3 s the aircraft measures the elevation with the standard deviation fixed to 15 m. 100 Monte Carlo trials have been performed. In averaging the results of the filters (when they converge), we compute the RMSE (Root Mean Square Error) 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 filter [15].

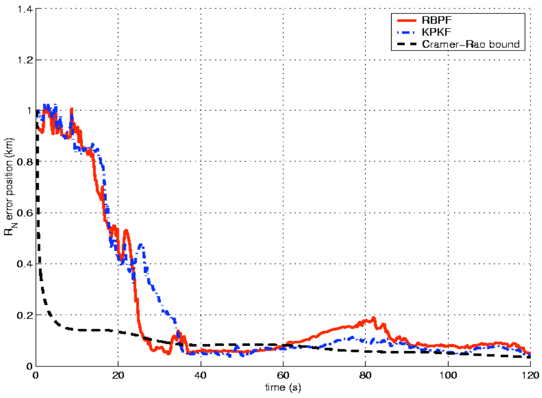

Figure 2: RMSE for the KPKF and for the RBPF and the PCRB for the R_N error position (km)

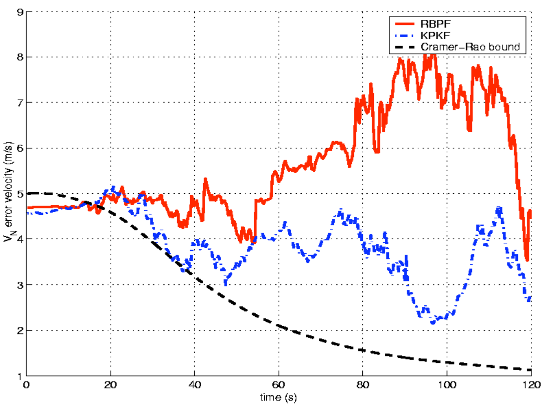

Figure 3: RMSE for the KPKF and for the RBPF and the PCRB for the V_N error velocity (m/s)

Results for the 2 filters are shown on Fig.2 and Fig.3 for R_N position and V_N velocity. For each trial, and for each measurement time, the aircraft state is estimated by the weighted mean of the particle cloud. The RPBF has yield 8 divergences (out of 100) while the KPKF only 2 divergences. We call divergence when the state estimate is consecutively 5 times out of the 99% confidence ellipsoid given by the PCRB. Both filters have good performances, the RMSE approach the PCRB (except for the V_N error for the RBPF). Other simulations with a larger uncertainty initial zone have been performed (not shown here). The KPKF also has much less divergences than the RBPF.

4. CONCLUSION

A new particle filter which incorporates the features of the Kalman filter is described. It is based on a representation of the conditional density as a kernel approximation. This allows local linearizations. An other originality of the filter is the partial resampling which reduces the Monte Carlo fluctuations. The number of resamplings is significantly reduced. Simulations in the difficult context of terrain navigation show that the KPKF outperforms usual particle filters like RBPF.

REFERENCES

[1] D. O. J. Benson, "A comparison of two approaches to pure-inertial and doppler-inertial error analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-7, pp. 447–455, July 1975.

[2] N. Bergman, "Recursive bayesian estimation, navigation and tracking applications," Ph. D. Thesis, Linköping University, Sweden, May 1999.

[3] G. Casella and C. P. Robert, "Rao blackwellisation of sampling schemes," *Biometrika*, pp. 81–94, 1996.

[4] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte-Carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[5] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Proceedings of the IEE*, vol. 140, no. 2, pp. 107–113, 1993.

[6] M. Hürzeler and H. R. Künsch, "Monte-Carlo approximations for general state space models," *Journal of Computational and graphical Statistics*, vol. 7, no. 2, pp. 175–193, 1998.

[7] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[8] J. S. Liu and R. Chen, "Sequential Monte-Carlo methods for dynamics systems," *J. Amer. Statist. Assoc.*, vol. 93, 1998.

[9] P. D. Moral, "Nonlinear filtering: interacting particle solution," *Markov Processes and Related Fields*, vol. 2, no. 4, pp. 555–580, 1996.

[10] C. Musso, N. Oudjane, and F. Legland, "Improving regularized particle filter," in *Sequential Monte-Carlo Method and Practice*, N. d. F. A. Doucet and N. Gordon, Eds. New-York: Springer, Jan. 2001, pp. 247–271.

[11] P. J. Nordlund, "Recursive estimation of three dimensional aircraft position using terrain aid positioning," in *Proceeding of the ICASSP Conference*, Orlando, U.S.A, May 2002.

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

[13] D. T. Pham, K. Dahia, and C. Musso, "A Kalman-particle kernel filter and its application to terrain navigation," in *Proceeding of the Fusion 2003 Conference*, Cairns, Australia, July 2003.

[14] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986.

[15] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posteriori Cramer-Rao bounds for dicrete-time nonlinear filtering," *IEEE Trans. Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.