PARTICLE FILTERING FOR NON-LINEAR SENSOR FUSION: APPLICATION TO TERRAIN-AIDED NAVIGATION

M. Flament, J-N. Lacave
MBDA, France

M-E. Davoust, G. Fleury
École Supérieure d'Électricité, France

Terrain aided navigation is a technique based on the association of an inertial navigation unit, a radio-altimeter measuring the distance from the aircraft to the ground and a digital terrain elevation map. The radio-altimeter measurements provide indirect information on the aircraft position that can be used to fix the inertial position, speed and attitude drift errors. This technique achieves a high accuracy navigation along the flight. This operation corresponds to a non-linear estimation problem. The non-linear property is induced by the terrain surface shape, which is a major difficulty for filtering algorithms, especially when large initial horizontal position errors and/or mountainous areas are considered.

Over the last three decades, several algorithms have been developed. Batch-oriented algorithms, like TERCOM, are based on the correlation between the measured terrain-height profile and the digital map. Another class of algorithm is derived from an Extended Kalman Filter. The SITAN algorithm is an example. More recent algorithms (TERPROM, SPARTAN) mix both approaches.

In this work, we focus on an alternative class of algorithms called particle filters [1] [2] [3]. Particle filters solve the estimation problem, in a Bayesian framework, by approximating the state probability density function by a set of weighted samples. We have developed two terrain-aided navigation algorithms based on the particle approximation. The first one is designed to cope with large initial position error (several kilometers). Thus, it can be used as an alternative to batch-algorithms for the position acquisition phase. Our second particle filter is based on the *Gaussian particle filter*, and offers an alternative to the Extended Kalman Filter approach. Both algorithms have been implemented and evaluated on simulated data, in comparison to grid-based methods and EKF algorithm.

The Bayesian approach & Particle filtering

Basically, terrain-aided navigation consists in combining the prior knowledge of the aircraft state given by the inertial system with the additional position-related information from terrain height measurements. The *Bayesian framework* provides the theoretical tools to carry out this operation in an optimal way. In particular, this framework is well adapted to the recursive estimation of a dynamic state model given independent observations [4].

Let's denote $\mathbf{x}_k$ the system state at time k. In the case of terrain-aided navigation, the state represents the inertial navigation errors. At a minimum, it includes position errors, but it may also be completed by speed and attitude errors. The state follows an evolution process described by a recursive equation:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad (1)$$

with $\mathbf{u}_k$ the evolution noise. In the Bayesian framework, an equivalent description of this process is given by the transition probability density function $p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k)$ which analytical expression can be straightforwardly drawn from f and the noise model.

The radio-altimeter measurement is modeled as:

$$m_k = z_k - h(l_k, L_k) + v_k \quad (2)$$

Where l_k, L_k and z_k are respectively the latitude, longitude, altitude of the aircraft. The function h stands for the terrain elevation model. The noise v_k sums up the measurements errors. Once again, this model can be equivalently described by a probability density function $p(m_k \,|\, \mathbf{x}_k)$, called likelihood.

In the Bayesian framework, the knowledge of the state $\mathbf{x}_k$ given all the past observations $\mathbf{M}_k=[m_1 \ldots m_k]$ is given by the posterior probability density function $p(\mathbf{x}_k \,|\, \mathbf{M}_k)$. This function can be recursively updated through the following equations:

$$p(\mathbf{x}_{k+1} | \mathbf{M}_k) = \int p(\mathbf{x}_{k+1} | \mathbf{x}_k)\, p(\mathbf{x}_k | \mathbf{M}_k)\, d\mathbf{x}_k \quad (3)$$

$$p(\mathbf{x}_{k+1} | \mathbf{M}_{k+1}) = \frac{p(m_{k+1} | \mathbf{x}_{k+1})\, p(\mathbf{x}_{k+1} | \mathbf{M}_k)}{p(m_{k+1} | \mathbf{M}_k)} \quad (4)$$

Subsequently, several state estimates can be build from the density function $p(\mathbf{x}_k \,|\, \mathbf{M}_k)$. A very classical choice *is minimum mean square error* (MMSE) estimate:

$$\hat{\mathbf{x}}_k^{\mathrm{MMSE}} = E_{p(\mathbf{x}_k | \mathbf{M}_k)}\left[\mathbf{x}_k\right]$$

This general scheme of recursive estimation defines the class of *Bayesian filters*. One of the well-known member of this class is the Kalman filter, which equations are directly drawn from *(3)* and *(4)* under the assumptions of linear models and additive Gaussian noises. Particle filters lies in the same framework. In addition, they provide the capability to handle non-linear and / or non-Gaussian problems. Indeed, equations *(3)* and *(4)* are analytically intractable in general case.

Therefore, a general approach for non-linear filtering is to choose an approximation of $p(\mathbf{x}_k \,|\, \mathbf{M}_k)$ that can be easily propagated through *(3)* and *(4)*. For particle filters, this approximation is based on a finite set of weighted samples called particles. More precisely, a particle consists in the association of a state vector $\mathbf{x}_k^{(i)}$ and a weight $w_k^{(i)}$. The particles are distributed into the state space and accordingly weighted so that the local density of weight around the state $\mathbf{x}_k$ is proportional to $p(\mathbf{x}_k \,|\, \mathbf{M}_k)$.

One of the attracting aspects of the particle filters is that the intractable update equations correspond to simple operations on the particle set:

- *Propagation* (also called *sampling*) step:

  Individual particle state is updated according to the state evolution model *(1)*. The noise $\mathbf{u}_k$ is randomly sampled for each particle.

- *Weight update* step:

  For each particle, a measurement prediction given the particle state is computed. The likelihood $p(m_{k+1} \,|\, \mathbf{x}_{k+1}^{(i)})$ is drawn from the difference between the predicted measurement and the actual observation m_k. Finally, weights are updated according to:

  $$w_{k+1}^{(i)} = p(m_{k+1} | \mathbf{x}_{k+1}^{(i)})\, w_k^{(i)}$$

That procedure ensures that the particle set remains a valid approximation of the true probability density $p(\mathbf{x}_k \mid \mathbf{M}_k)$ at any time k. Nevertheless, as observations are accumulated, the total amount of weight concentrates on only few particles and a majority of particles have a negligible weight. Thus, an additional operation called *resampling* is needed. It consists in discarding low weighted particles and multiplying the others ones.

In addition, the MMSE estimate of the state is computed as the weighted mean of the particle states:

$$\hat{\mathbf{x}}_k^{\mathrm{MMSE}} = \sum_{i=1}^{N} w_k^{(i)} \mathbf{x}_k^{(i)}$$

Finally, the overview the whole algorithm structure is presented on fig. 1.

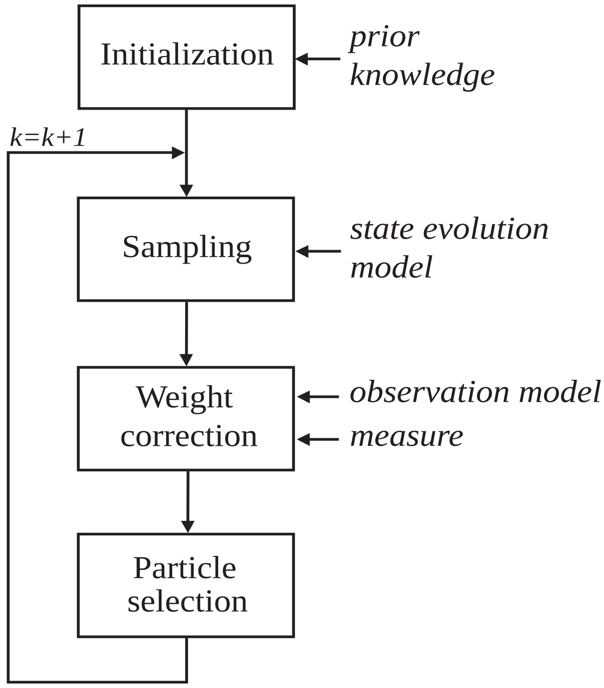

Fig 1. Particle filter algorithm

Practical implementation of particle filters for terrain-aided navigation

The particle filter algorithm, as described in the previous section, is more a general scheme than a ready-to-use algorithm. The following sections will be dedicated to the adaptation of the algorithm to the specific issues raised by the terrain-aided navigation filtering problem. Therefore, efforts have been aimed in two directions: firstly, improving the algorithm efficiency and reliability, and secondly, pointing out the interests of using particle filter over traditional grid-based or EKF-based methods.

Rao-Blackwellization procedure

An important aspect of particle filter design is the number of particles that ensure an correct probability density approximation. This aspect is strongly related to the dimension of the state space: the number of required particles grows exponentially with the dimension. As a result, particle filters cannot handle high-dimension state space with a number of particles compatible with a real-time, embedded implementation. For TAN, this problem can be solved by taking advantage of the particular structure of the problem. Indeed, the non-linear nature of the problem is clustered in the relationship between the measurement and the horizontal position. This allows a factorization of $p(\mathbf{x}_k \mid \mathbf{M}_k)$ to the form:

$$p(\mathbf{x}_k | \mathbf{M}_k) = p(\mathbf{x}_k^H | \mathbf{M}_k)\, p(\mathbf{x}_k^N | \mathbf{x}_k^H, \mathbf{M}_k)$$

where $[\mathbf{x}_k^H, \mathbf{x}_k^N]$ is a partition of the state vector between horizontal position and the other variable. The key point is that the determination of $p(\mathbf{x}_k^N \mid \mathbf{x}_k^H, \mathbf{M}_k)$ is a linear Gaussian problem (under the assumption that the measurement noise is Gaussian) and can be solved analytically with a Kalman filter. The non-Gaussian remaining part $p(\mathbf{x}_k^H \mid \mathbf{M}_k)$ is approximated by a set of particles which are sampled in the horizontal plane only. This procedure is called Rao-Blackwellisation. Further developments can be found in [2].

Particle filtering for position fix

In this section, particle filtering is applied to a situation of large initial position errors. In such case, the terrain variations over the initial uncertainty area are too important for a proper

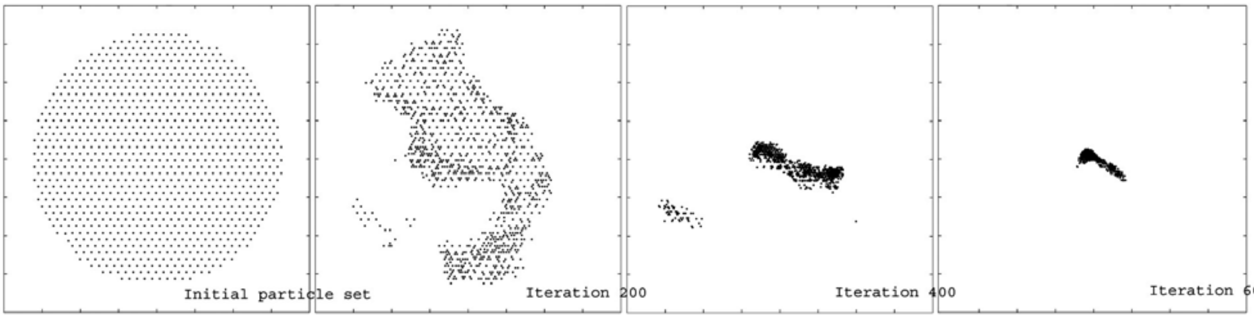

Fig. 2. Evolution of the particle set in the horizontal plane

terrain linearization. Such a scenario occurs after a long inertial-only navigation cruise phase.

The proposed algorithm aimed at jointly estimating position and speed on each axis, considering the following state model (for one axis):

$$\begin{bmatrix} p_{k+1} \\ \dot{p}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ \dot{p}_k \end{bmatrix} + \begin{bmatrix} 0 \\ u_{acc} \end{bmatrix} \quad (5)$$

where u_{acc} is an acceleration noise, and ΔT the iteration timestep. A Rao-Blackwellisation is applied to partition the state space between horizontal variables and vertical ones: a Kalman filter is attached to each particle and estimates the vertical position and speed given the particle trajectory in the horizontal plane.

Here is the description of each filter step:

Initialisation step:

(a) The prior state density is assumed to be Gaussian. Particles are laid uniformly over a circular area that correspond to a 3σ uncertainty boundary (see fig. 2). Their weights are ajusted to fit a Gaussian distribution.

(b) The vertical channel Kalman filters are initialized according to the prior information on the vertical channel.

Sampling step:

(a) The horizontal state is updated according to *(5)*. An acceleration noise value is randomly sampled for each particle.

(b) A prediction step is performed on Kalman filters, according to *(5)*.

Weight update step:

(a) A measurement prediction $m^{(i)}$ is computed for each particle. Kalman filter are updated according to the innovation $\xi^{(i)} = m_k - m^{(i)}$.

(b) Particle weight are updated according to the innovation likelihood:

$$w_{k+1}^{(i)} = L(\xi^{(i)}) \cdot w_k^{(i)}$$

The fig. 2 presents the evolution of the particle set as the filter converges. Such a particle filtering algorithm is comparable to grid-based method. Indeed, if a null horizontal speed is assigned to particles and if the particle selection step is switch off, the algorithm is very close to a fixed-grid method. With the particle selection step, it behaves as an adaptive grid method. Hence, the main contribution of particle filter over grid-based methods is the possibility to jointly estimate 3D position and speed.

Particle filtering for tracking

As soon as the position fix is achieved with an acceptable accuracy, a tracking algorithm can be launched to track the INS errors with an elaborated drift model. In addition to position and speed errors, this model may include attitude errors, accelerometer bias and gyro-meter drift rates. This task is classically devoted to an extended Kalman filter, which efficiently handles high dimension state evolution models. This approach is followed in the SITAN algorithm for example. In essence, the performances of an EKF-based filter for terrain-aided navigation are limited by the relevance of the local linear approximation of the terrain profile. Consequently, a severe risk of failure appears in case of large horizontal position errors.

By contrast, particle filters do not rely on any terrain approximation, since terrain elevations are evaluated locally for each particle. However, particle filters do not handle efficiently high state space dimension.

The proposed algorithm is an effort to conciliate the attributes of interest of each approach. The basic structure of this filter is derived from the Kalman filter: the filtering densities $p(\mathbf{x}_k | \mathbf{M}_k)$ are assumed to be Gaussian and the prediction step is identical to a Kalman filter. The key idea of the proposed algorithm consists in substituting the Kalman update step with a particle-based procedure, so as to handle stronger non-linearities.

The algorithm has been implemented using a state model which includes position, speed and attitude (9 dimensions). Here is its detailed structure:

Prediction step: (Identical to a Kalman filter)

$$\hat{\mathbf{x}}_{k+} = \mathbf{F}_k \hat{\mathbf{x}}_k$$

$$\mathbf{P}_{k+} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k$$

Update step:

(a) *sampling*: Particles are sampled in the horizontal plane according to the Gaussian prior distribution $p(\mathbf{x}_{k+1}^H / \mathbf{M}_k)$:

$$\mathbf{x}_{k+}^{H,(i)} = \hat{\mathbf{x}}_{k+}^H + \mathbf{u}^{(i)}$$

where $\mathbf{u}^{(i)}$ is a random Gaussian vector with covariance $\mathbf{P}_{k+1/k}^H$.

The distribution $p(\mathbf{x}_{k+1}^N | \mathbf{x}^{H,(i)}, \mathbf{M}_k)$ is also Gaussian, with mean and covariance given:

$$\hat{\mathbf{x}}_{k+}^{N,(i)} = \hat{\mathbf{x}}_{k+}^H + M(P_{k+})^{-1}(\hat{\mathbf{x}}_{k+}^{H,(i)} - \hat{\mathbf{x}}_{k+}^H)$$

$$\mathbf{P}_{k+}^{N,(i)} = \mathbf{P}_{k+}^N - \mathbf{M}(\mathbf{P}_{k+})^{-1}\mathbf{M}^T$$

(b) *Kalman update and weighting*: The determination of $p(\mathbf{x}_{k+1}^N | \mathbf{x}^{H,(i)}, \mathbf{M}_{k+1})$ from $p(\mathbf{x}_{k+1}^N | \mathbf{x}^{H,(i)}, \mathbf{M}_k)$ and the new measure m_k is a linear Gaussian problem. A Kalman update step is performed for each particle to obtain the updated mean $\hat{\mathbf{x}}_{k+1}^{N,(i)}$ and covariance $\mathbf{P}_{k+1}^{N,(i)}$. In addition, particle weights are updated from the likelihood of the innovation $\xi^{(i)}$ given by the Kalman filter:

$$w_{k+1}^{(i)} = L(\xi^{(i)}) \cdot w_k^{(i)}$$

(c) *Estimation*: the mean $\hat{\mathbf{x}}_{k+1}$ and covariance $\mathbf{P}_{k+1}$ of the posterior distribution is estimated from the particle set.

The fig. 3 presents the particle set, sampled from the prior distribution, then weight according to their likelihood (blue-green-red color), and the corresponding posterior distribution.

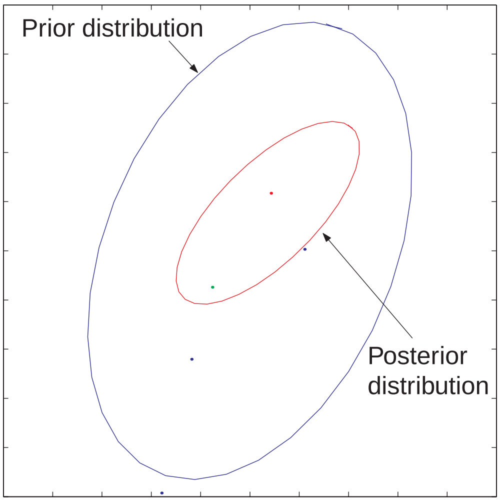

Fig.3. weighed particle set

Simulations and results

A generic model of *strap-down* inertial navigation system is used to generate inertial data from a simulated aircraft trajectory. The radar-altimeter data are generated from the trajectory and a terrain map. Radar-altimeters errors are considered to be a Gaussian noise.

The fig. 4 presents the terrain map used for the evaluation, with two test trajectories. The first one (scenario1) corresponds to a hilly terrain, and thus the measurements are very informative. On the contrary, the second one (scenario 2) exhibits less terrain variations and a longer measurement sequence is needed.

The evaluation criteria is the residual root mean square error for horizontal position, evaluated on the basis of 50 runs.

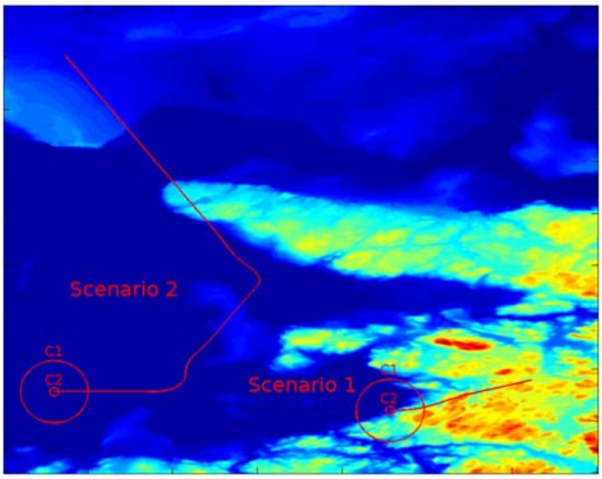

Fig.4. Terrain map and scenarios

Particle filtering versus grid methods

This comparison involves the particle filter designed for position fix and a grid-based method. The grid-based method operates on a fixed horizontal grid, which corresponds to the initial particle set (fig. 2). The initial uncertainty area is denoted as “C1” on fig. 4.

The results for both scenarios are presented fig. 5 and fig. 6. The convergence phases for both filters are very similar. However, the speed drift degrades the grid method estimate over time. That phenomenon is clear when long sequences are considered (scenario 2). On the contrary, as the particle filter jointly estimates speed errors, it is usable on long sequences.

Particle filtering versus EKF algorithm

This comparison involves the Gaussian particle filter designed for tracking and a standard extended Kalman filter. The initial uncertainty area is denoted as “C2” on fig. 4.

The results for both scenarios are presented fig. 7 and fig. 8. On hilly terrain (scenario 1), the particle filter exhibits a faster convergence rate, and the final precision is slightly better. In addition, 10 divergences cases have been observed for the EKF, and only 3 for the particle filter. In addition, the results for scenario 2 illustrate the capability of the particle filter to extract information from small terrain variations.

Conclusion

From this study, several conclusions can be drawn concerning the relevance of using particle filters for terrain-aided navigation. For a practical point of view, particle filters are a worth alternative to grid-based methods in situations were the drift induced by speed errors cannot be neglected over the measurement time lag. Such situations occur in case of poor featured terrain that requires a long integration time to obtain a precise position fix. In other aspects, the Gaussian particle filter can also be used as a direct replacement of an EKF. In that application, the main interests are a faster convergence and a better robustness. More generally, particle estimation is a very flexible framework: various algorithms can be drawn from the general principles and targeted to a specific context of use.

Acknowledgement

This study has been achieved in the framework of a Ph.D. work, supported by MBDA France.

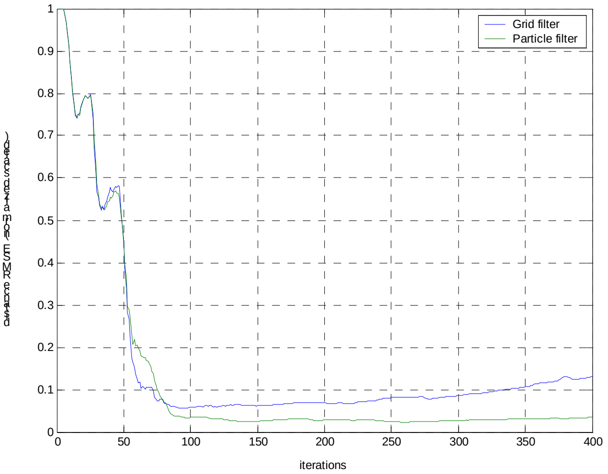

Fig. 5. Scenario 1 – Position fix

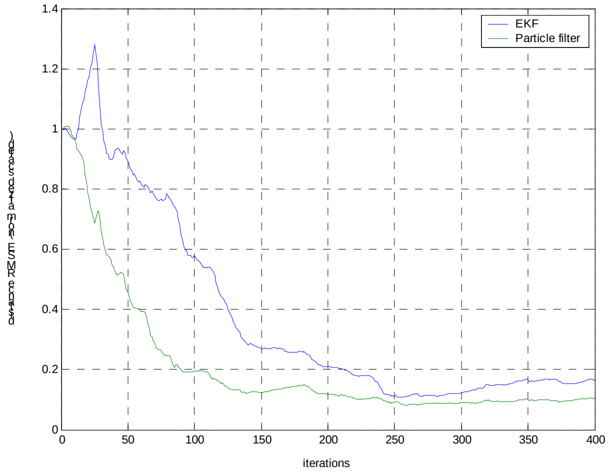

Fig. 7. Scenario 1 - Tracking

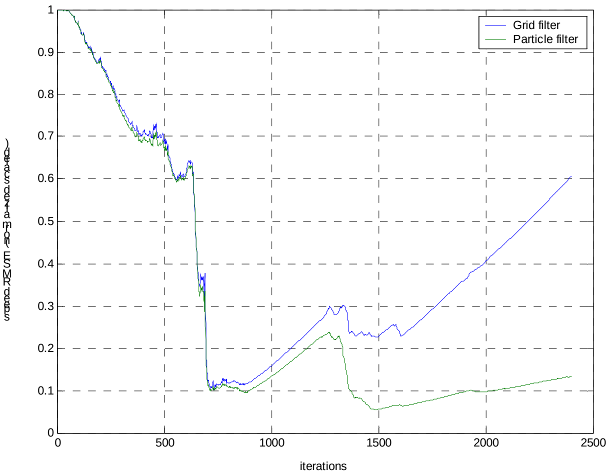

Fig. 6. Scenario 2 – Position fix

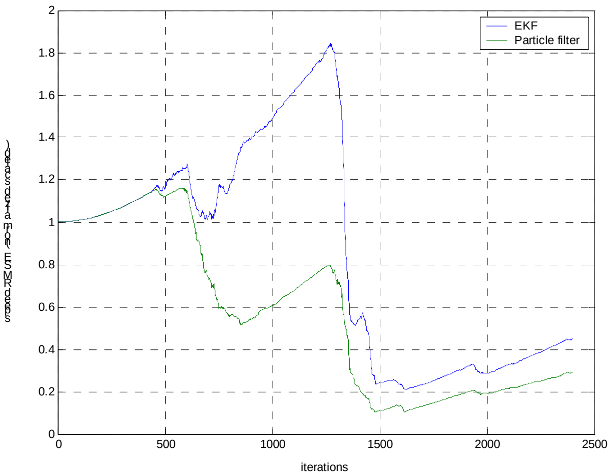

Fig. 8. Scenario 2 - Tracking

References

[1] C. Musso and N. Oudjane. Recent particle filter applied to terrain navigation. In *Proceedings of the Third International Conference on Information Fusion*, vol. 2, pp. 26-33, July 2000.

[2] P. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation.* PhD thesis, Division of Automatic Control, Departement of Electrical Engineering, Linkoping University, Sweden, 2002.

[3] Mathieu Flament, Gilles Fleury and Marie-Eve Davoust Particle Filter and Gaussian-Mixture Filter Evaluation for Terrain-Aided Navigation, In *Proceedings of European Signal Processing Conference*, pp. 630-634, September 2004.

[4] N. Bergman, L. Ljung, and F. Gustafsson. *Terrain navigation using Bayesian statistics.* Technical Report LITH-ISY-R-2139, Departement of Electrical Engineering, Linkoping University, Sweden, 1999.

[5] Jayesh H. Kotecha and Petar M. Djuric. Gaussian Particle Filtering, *IEEE Transactions on Signal Processing*, vol. 51, no. 10, October 2003.