PARTICLE FILTER AND GAUSSIAN-MIXTURE FILTER EFFICIENCY EVALUATION FOR TERRAIN-AIDED NAVIGATION

Mathieu Flament[†] - *Gilles Fleury*[‡] - *Marie-Ève Davoust*[‡]

[†] MBDA France
37 avenue Louis Breguet BP150
F-78141 Vélizy-Villacoublay Cedex, France
mathieu.flament@mbda.fr

[‡] École Supérieure d'Électricité
3 rue Joliot-Curie, Plateau de Moulon
F-91192 Gif-sur-Yvette Cedex, France
gilles.fleury@supelec.fr, marie-eve.davoust@supelec.fr

ABSTRACT

Terrain-aided navigation is a method relying on a digital terrain elevation database and radar-altimeter measurements and can be applied to manned or unmanned aircrafts. Associated with an inertial navigation system, terrain-aided navigation provides an accurate estimation of position. Since the aircraft state estimation implies non-linear filtering, the computational load of terrain-aided navigation algorithms is generally high. Hence, for real-time implementation, non-linear filters should be designed to achieve maximum performances with limited resources. In this work, we focus on particle filter and Gaussian-mixture filter which are two classical approaches to solve non-linear problems in a Bayesian framework. We describe the two algorithms and compare their performances on various terrain topographies. These simulations highlight that the Gaussian-mixture filter achieves better performances and reliability, in a situation where the filter design aims at reducing computational requirements.

1. INTRODUCTION

With the recent development of autonomous aircrafts (unmanned aerial vehicles, long-range cruise missiles), navigation reliability and accuracy are key parameters for mission achievement. Traditional autonomous navigation systems are based on inertial measurements. On-board accelerometers and gyro-meters sense the aircraft movements. From these measurements, a navigation computer continuously updates the estimated position, speed and attitude of the aircraft. Inertial navigation systems (INS) are fully autonomous and require no external aid. Unfortunately, initialization errors and cumulated sensor errors lead to increasing uncertainties on INS outputs. So as to bound INS errors, additional position-related measurements are needed. Various systems can be used in conjunction with inertial navigation systems: barometric altimeter, GPS receiver, optical or radar sensors. For an overview, see [1].

In this work, we focus on terrain-aided navigation (TAN). Terrain-aided navigation encompasses all methods based on a comparison between a terrain elevation measurement and an on-board digital terrain elevation database. A thorough introduction can be found in [2]. A simple technical solution consists in measuring the distance between the aircraft and the ground (ground clearance) with a radar-altimeter (see figure 1). This distance corresponds to the difference between the aircraft altitude and the elevation of the area flown over (stored in the elevation database). Observations are made regularly along the flight path.

The radar-altimeter provides an additional, indirect information on aircraft position. The sensor-fusion problem consists in filtering inertial sensors and radar-altimeter outputs to achieve an optimal estimation of the aircraft state.

The optimal sensor-fusion problem can be solved in a Bayesian framework [3]. Unfortunately, the optimal Bayesian filter is intractable. The major difficulty is raised by the non-linear relation between the radar measurement and the aircraft position. A classical approach is based on local terrain linearization and Extended Kalman filter. On hilly terrains, local linearization hypothesis is only valid over small range, depending on the terrain roughness. Therefore, Extended Kalman filter based methods do not perform well and must be carefully initialized to avoid filter divergence. Besides, for a highly non-linear problem, single Gaussian approximation of the posterior state density is not satisfactory.

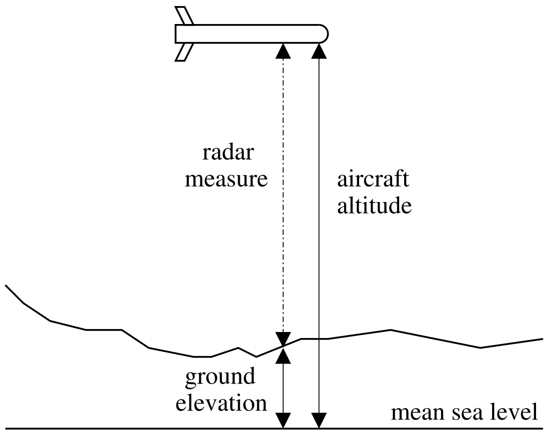

Figure 1: Radar-altimeter measurement

Furthermore, the disparity of natural terrain topography becomes an additional difficulty for robust filter design and tuning. On smooth terrains, the observability on horizontal position is low. On rough terrains, similarities between several areas can result in ambiguities on estimated position.

Several solutions have been developed to overcome these difficulties. *Particle filtering* has been successfully applied to terrain-aided navigation [4] [5]. The particle filter is based on Monte Carlo sampling, and thus requires heavy computations. The *Gaussian-mixture filter* is also suitable for this problem [6]. We propose to use a Gaussian-mixture filter with an updating step inspired from Unscented Kalman filter algorithm [7]. Since the Gaussian-mixture filter is not based on Monte Carlo methods, we expect a better efficiency at equal computational cost. Needed, for a real-time implementation, where computational resources are limited, efficiency is a key criterion for filter design. In the terrain-aided navigation context, the filter should be usable for a large variety of terrain (smooth, hilly or mountainous), it should converge as quickly as possible and should avoid divergence. In this work, we compare the particle filter and the Gaussian-mixture filter on that set of evaluation criteria.

In section 2, INS drift and measurement models are presented, followed by the optimal Bayesian filter equations. The particle filter and Gaussian-mixture filter algorithms are described in section 3. In section 4, the two algorithms are compared in various situations, and the results will be discussed in the last section.

2. THE BAYESIAN APPROACH

2.1 INS/radar-altimeter hybrid navigation principle

First, a purely inertial aircraft state estimation is computed from accelerometer and gyro-meter measurements. Then the sensor-fusion filter estimates the difference between the actual state and that inertial estimation (i.e. the INS drift). Finally, the INS drift estimate is used to correct the INS output (see figure 2).

This approach has several advantages over a direct filtering scheme. Firstly, the radar-altimeter data (few hertz rate) can be treated asynchronously from inertial sensors measurements (several hundreds of hertz). Secondly, the INS drift dynamics can be represented by a linear model. Finally, the inertial estimate remains available and uncorrupted in case of filter divergence or radar-altimeter failure.

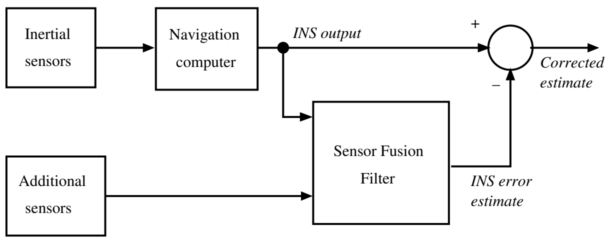

Figure 2: Sensor-fusion filter

2.2 INS drift model

The filter state contains INS position errors (in latitude, longitude and altitude), speed errors and attitude errors (roll, pitch and yaw). This is a minimal error state for a 6 degrees of freedom vehicle. The state can be extended with other estimates of interest (for example, accelerometer and gyro-meter bias). We denote $\mathbf{x}_k$ the filter state at time k. A generic linear model for INS drift can be written as:

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + G_k \mathbf{u}_k \tag{1}$$

where $\mathbf{u}_k$ is the noise vector (on accelerometer and gyro-meter outputs), $\mathbf{u}_k$ is assumed to be a zero mean, stationary Gaussian noise. The matrix F_k and G_k are deduced from inertial navigation equations. Their expressions depends on the inertial system technology. In this work, we use a *strap-down* inertial system drift model. Further discussions on INS drift models can be found in [8].

2.3 Measurement model

The measurement equation is given:

$$y_k = z_k - h(l_k, L_k) + v_k \tag{2}$$

where l_k, L_k and z_k are respectively the longitude, the latitude and the altitude of the aircraft. The function h stands for the terrain profile stored on-board. For the experiments, we used 3 arc-second elevation maps (sample points are spaced out 1/1200 degree on latitude and longitude). The terrain elevation is reconstructed by bilinear interpolation. The noise v_k represents the measurement error. It includes the radar-altimeter and map errors (difference between actual terrain and the profile reconstructed from the digital map). The choice of a model for the measurement noise is not trivial. Indeed, radar-altimeter error sources are multiple and depend on many factors (distance from the ground, terrain roughness, type of vegetation). Moreover, errors are time correlated (the closer the measurement points, the stronger the correlation). In our work, we consider v_k as a Gaussian white noise. This hypothesis can be considered as a rough simplification. However, as we aim at comparing particle filter and Gaussian-sum filter performances, this simplification is acceptable. Further investigations could be done with experimental data.

2.4 Bayesian solution

Considering the state space model (1) and the measurement equation (2), the solution of optimal filter in the Bayesian framework is achieved in two steps:

- Recursively compute the posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_k)$ of state space vector given observations:

$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}) d\mathbf{x}_{k-1} \tag{3a}$$

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \frac{p(y_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{\int p(y_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Y}_{k-1}) d\mathbf{x}_k} \tag{3b}$$

  where $\mathbf{Y}_k = [y_0, \ldots, y_k]$. The expressions of $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(y_k|\mathbf{x}_k)$ are straightforwardly drawn from (1) and (2) and noise density models.
- Compute a state estimator. The minimum mean square error estimator (MMSE) is a classical choice:

$$\hat{\mathbf{x}}_k^{MMSE} = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{Y}_k) d\mathbf{x}_k$$

3. FILTER DESIGN

Multidimensional integrals that appear in (3a) and (3b) are analytically intractable in general case, because of the non-linearities introduced by the terrain profile. A general approach for nonlinear filtering is to choose an approximation for $p(\mathbf{x}_k|\mathbf{Y}_k)$ and to propagate that approximation through (3a) and (3b). Non-linear filters can be classified from the nature of this approximation. Most common non-linear filters for terrain-aided navigation are:

- the *point-mass filter* uses a fixed (or adaptive) discretization of the state space (a "grid"); $p(\mathbf{x}_k|\mathbf{Y}_k)$ is recursively computed for each node of the discrete grid.
- in the *Gaussian-mixture filter*, $p(\mathbf{x}_k|\mathbf{Y}_k)$ is approximated by a weighted Gaussian mixture.
- the *particle filter* propagates a set of weighted samples distributed according to $p(\mathbf{x}_k|\mathbf{Y}_k)$.

In the following section, we present the particle filter and the Gaussian-mixture filter algorithms.

3.1 Particle filter algorithm

Particle filter theory is explained in [9] and [10]. The principle of particle filter is to use a set of weighted samples, called particles, which represent a Monte-Carlo approximation of $p(\mathbf{x}_k|\mathbf{Y}_k)$. Each step of the filter consists in updating particle states and weights to keep this approximation valid (in a stochastic meaning). The generic particle filter algorithm is presented figure 3.

The choice of the so-called *proposal distribution* $q(\mathbf{x}_k|\mathbf{X}_{k-1}^i, \mathbf{Y}_k)$ - the distribution from which particles are sampled at each iteration - is a crucial point of particle filter design. More precisely, particle filters are affected by the *particle degeneracy* phenomenon [9]: the variance of particle weights is increasing over time and periodic resampling operations are needed. The choice of a relevant proposal distribution can minimize the degeneracy process [10]. In this work, we evaluate two different proposal distributions:

- $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ which is the simpler and most common distribution. Particles follow a random walk independently from the measurements. This choice requires light computations but leads to a severe variance growth, as $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is far from the optimal proposal distribution.
- An interesting method for devising a proposal distribution closed to the optimal one is to use local linearization around each particle and to incorporate the last observation with a Kalman filter associated to each particle. The computational cost per particle is higher, but the number of particles can be significantly reduced. This is the principle of the *Extended Kalman*

particle filter and the *Unscented particle filter* [11]. The latter is based on the *Unscented Kalman filter* [7], an attractive alternative to EKF for highly non-linear problems.

3.2 Gaussian-mixture filter algorithm

This method was introduced in [6]. A recent application to tracking is presented in [12]. The state density $p(\mathbf{x}_k|\mathbf{Y}_k)$ is here approximated by a weighted mixture of Gaussians:

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \sum_{i=1}^{N} \omega_k^i \mathcal{N}_{\mathbf{x}_k}(\mu_k^i, P_k^i) \tag{4}$$

The mixture parameters $\{\omega_k^i, \mu_k^i, P_k^i\}_{i=1..N}$ are recursively updated with an Unscented Kalman filter bank. The algorithm is described in figure 4.

Contrary to the particle filter, the Gaussian-mixture filter do not rely on a sampling procedure. Moreover, process and measurement noises must be additive and Gaussian. This limitation does not exist for particle filter. Nevertheless, Unscented particle filter and Gaussian-mixture filter algorithms are closely related and their computational loads are equivalent.

4. SIMULATIONS

In this work, we implement and evaluate 3 distinct filters:

- A *basic particle filter*: ($p(x_k|x_{k-1})$ is chosen as the *proposal distribution*). This filter is initialized with 5000 particles. We add a regularization step [10] after the resampling procedure, so as to improve the state space exploration.
- An *Unscented particle filter* with 1000 particles.
- A *Gaussian-mixture filter*. The initial mixture is composed from N=500 Gaussian terms :

$$p(\mathbf{x}_0) = \sum_{i=1}^{N} \frac{1}{N} \mathcal{N}_{\mathbf{x}_0}(\mu_0^i, P_0)$$

  The means μ_0^i are uniformly distributed over the initial search region and the covariance matrix P_0 is adjusted to obtain a smoothed density function.

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 terrain maps. The altimeter sample rate is 2Hz. A Gaussian noise is added to simulated output such as :

$$v_k \sim \mathcal{N}(0, 100m^2)$$

The simulation consists of 100 Monte-Carlo runs. For each run, a new initial INS error is randomly chosen.

To compare algorithm performances, we focus on the following criteria:

- *Robustness versus terrain topography*: each algorithm was evaluated on three terrain: a smooth one (terrain elevation between 0 and 110m), a hilly one (180m to 420m) and a mountainous one (660m to 2260m). Filter parameters are adjusted to achieve best performances on the hilly terrain (the "nominal" terrain). The table 1 presents the number of failed runs for 100 simulations. This test points out a significant degradation of the particle filter performances on smooth and mountainous terrains. Using a Gaussian-mixture filter, this effect is clearly minimized.
- *Computational load*: we run the simulation on a Pentium IV 1.8GHz. All filters are implemented in C++. The table 1 shows that the Gaussian-mixture filter achieves good performances for a reduced computational load.
- *Convergence speed*: the root mean square error (RMSE) of each filter over time is presented figure 6. The Gaussian-mixture filter is equivalent to the Unscented Kalman particle filter. The basic particle filter converges much slower. This is due to the regularization step, which adds an artificial noise on particle trajectories.

Initialization
Sample N particles from the initial distribution $p(\mathbf{x}_0)$:

$$\mathbf{x}_0^i \sim p(\mathbf{x}_0) \quad \text{and} \quad \omega_0^i = 1/N$$

For $k = 1, \cdots, \infty$

(1) *Sample particles from a proposal distribution*

$$\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{X}_{k-1}^i, \mathbf{Y}_k)$$

(2) *Update and normalize particle weights*

$$\omega_k^{i'} = \omega_{k-1}^i \frac{p(y_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{X}_{k-1}^i, \mathbf{Y}_k)} \quad \text{and} \quad \omega_k^i = \omega_k^{i'} / \sum_{i=1}^{N} \omega_k^{i'}$$

(3) *Resampling (optional)*
Discard/duplicate particles according to their weight to generate a new set of equally weighted particles.

(4) *Evaluate MMSE estimator*

$$\hat{\mathbf{x}}_k^{MMSE} = \sum_{i=1}^{N} \omega_k^i \mathbf{x}_k^i$$

Figure 3: Particle filter algorithm

Initialization
Choose ω_0^i, μ_0^i and P_0^i according to the initial distribution $p(\mathbf{x}_0)$:

$$p(\mathbf{x}_0) \approx \sum_{i=1}^{N} \omega_0^i \mathcal{N}_{\mathbf{x}_0}(\mu_0^i, P_0^i)$$

For $k = 1, \cdots, \infty$

(1) *Time update*

$$\mu_{k/k-1}^i = F_k \mu_{k-1}^i$$
$$P_{k/k-1}^i = F_k P_k^i F_k^\top + G_k Q_k G_k^\top$$

(2) *Measurement update*
Compute $\tilde{y}_k^i$, $P_{\mathbf{x}_k \tilde{y}_k}^i$ and $P_{\tilde{y}\tilde{y}}^i$ with Unscented transformation method (see [7]) and update mixture parameters:

$$K_k^i = P_{\mathbf{x}_k \tilde{y}_k}^i \left(P_{\tilde{y}\tilde{y}}^i\right)^{-1}$$
$$\mu_k^i = \mu_{k/k-1}^i + K_k^i \tilde{y}_k^i$$
$$P_k^i = P_{k/k-1}^i - K_k^i P_{\tilde{y}\tilde{y}}^i K_k^{i\,\top}$$

(3) *Weight update*

$$\omega^{i'} k = \omega_{k-1}^i \mathcal{N}_{\tilde{y}_k^i}(0, P_{\tilde{y}\tilde{y}}^i) \quad \text{and} \quad \omega_k^i = \omega_k^{i'} / \sum_{i=1}^{N} \omega_k^{i'}$$

(4) *Evaluate MMSE estimator*

$$\hat{\mathbf{x}}_k^{MMSE} = \sum_{i=1}^{N} \omega_k^i \mu_k^i$$

Figure 4: Gaussian-mixture filter algorithm

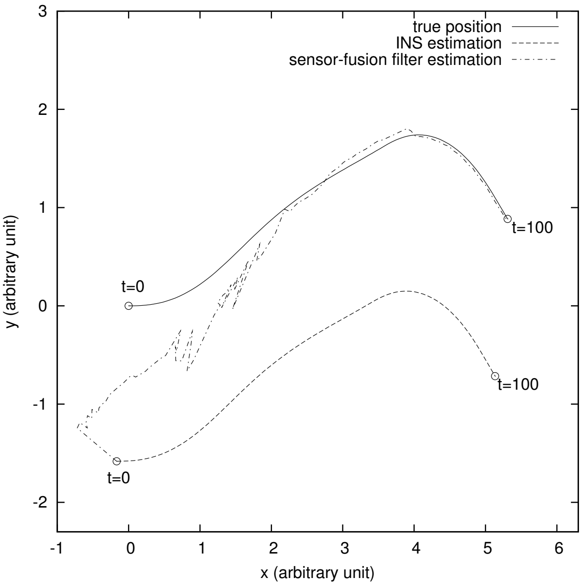

Figure 5: Convergence of the mean position estimate from a Gaussian-mixture filter (in the horizontal plane)

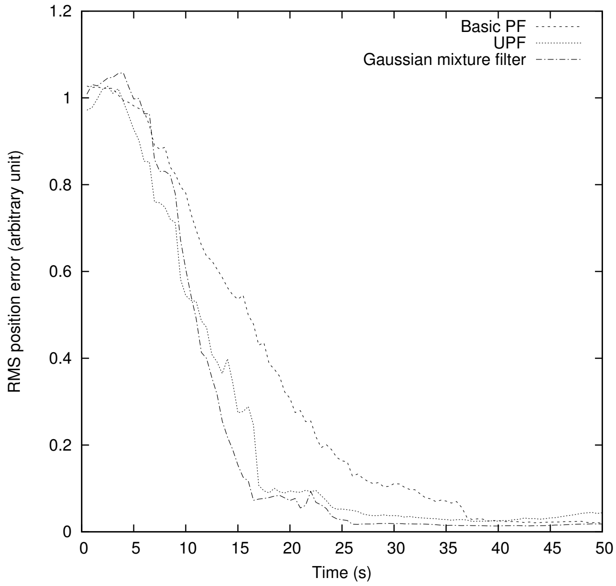

Figure 6: RMS error for basic particle filter (solid line), unscented Kalman particle filter (dashed line), and Gaussian-mixture filter (dotted line). Estimation based on 100 Monte-Carlo runs using the hilly terrain.

	Time	Failed runs		
	(s)	Smooth	Hilly	Mount.
Basic PF (N=5000)	20.4	68	20	57
UPF (N=1000)	23.2	59	18	35
GM-F (N=500)	10.1	35	9	24

Table 1: Robustness test results

5. CONCLUSION

In this work, we evaluate several non-linear filter designs for terrain-aided navigation, which differ from the nature of the state density approximation. The particle filter uses a discrete approximation and thus requires a large number of particles to ensure a consistent representation of the state density. Moreover, the simulations have shown that a particle filter with a simple *proposal distribution* is very sensitive to the terrain topography. To reduce the number of particles and improve filter performances, a refined *proposal distribution* is needed. In this purpose, the superiority of the Unscented particle filter has been illustrated. Nevertheless, the inherent stochastic nature of particle filters limits the particle number reduction. In case of limited computational resources, the Gaussian-mixture filter appears as an interesting alternative. In order to approximate the state density with a reduced set of parameters, the smooth Gaussian-mixture representation is more suitable than a discrete one. Indeed, the simulation highlights the benefit of using a Gaussian-mixture filter, which offers an overall performance improvement and enables a significant computational cost reduction.

REFERENCES

[1] H. Durrant-Whyte. A critical review of the state-of-the-art in autonomous land vehicle systems and technology. Technical Report SAND2001-3685, Sandia National Laboratories, 2001.

[2] Neil Priest. Terrain referenced navigation. In *Position, Location and Navigation Symposium Record*, pages 482–489. IEEE, March 1990.

[3] 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.

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

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

[6] D.L. Alpach and H.W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, August 1972.

[7] E. Wan and R. van der Merwe. The Unscented Kalman Filter for nonlinear estimation. In *Proceedings of Symposium 2000 on Adaptative Systems for Signal Processing, Communication and Control*. IEEE, 2000.

[8] P. Faurre. *Navigation Inertielle Optimale et Filtrage Statistique*. Dunod, 1971.

[9] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–88, February 2002.

[10] A. Doucet, S.J. Godsill, and C. Andrieu. On sequential simulation-based methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[11] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan. The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR 380, Departement of Engineering, University of Cambridge, 2000.

[12] V. Teullière. *Contribution au filtrage de Volterra et l'estimation particulaire - Application aux transmissions électroniques et acoustiques*. PhD thesis, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, 2000.