RECURSIVE ESTIMATION OF 3-DIMENSIONAL AIRCRAFT POSITION USING TERRAIN-AIDED POSITIONING

Per-Johan Nordlund and Fredrik Gustafsson

Department of Electrical Engineering
Linköping University
SE-58183 Linköping, Sweden
perno@isy.liu.se, fredrik@isy.liu.se

ABSTRACT

As a part of aircraft navigation, three-dimensional position must be computed continuously. For accuracy and reliability reasons, several sensors are integrated together, and here we are dealing with dead-reckoning integrated with terrain-aided positioning. Terrain-aided positioning suffers from severe nonlinear structure, meaning that we have to solve a nonlinear recursive Bayesian estimation problem. This is not possible to do exactly, but recursive Monte Carlo methods, also known as particle filters, provide a promising approximate solution.

To reduce the computational load of the normally rather computer intensive particle filter we present an algorithm which takes advantage of linear structure. The algorithm is based on a Rao-Blackwellisation technique, meaning that we marginalise the full conditional posterior density with respect to the linear part. The linear part of the state vector is estimated using multiple Kalman filters, and the particle filter is then used for the remaining part. Simulations show that the computational load is reduced significantly.

1. INTRODUCTION

The purpose of this paper is to provide optimal, yet tractable, methods for estimation of three-dimensional aircraft position (horizontal position and altitude), under the assumption that measurements are obtained from a dead-reckoning and a terrain-aided positioning system.

The integration of dead-reckoning with terrain-aided positioning is a nonlinear recursive Bayesian state estimation problem, nonlinear mainly due to the highly nonlinear nature of terrain-aided positioning and recursive because we need the estimates on-line. The classical approach to this type of problems is to use the extended Kalman filter. However, for systems with severe nonlinear and/or non-Gaussian structure the extended Kalman filter technique is not adequate. Another way to deal with nonlinear estimation problems is to use simulation based methods, also referred to as sequential Monte Carlo methods, or particle filters [1, 2, 3]. Although they do not suffer from the curse of dimensionality, in many cases a lot of particles are needed, especially when the dimension of the problem increases, making these methods computer intensive.

The outline of the report is as follows. In Section 2, the estimation problem is described in detail. Section 3 gives a brief introduction to particle filtering and Section 4 deals with methods for reducing the computational load. Section 5 provide simulation results. Finally, in Section 6 conclusions are drawn.

2. PROBLEM FORMULATION

The principle for a dead-reckoning system (e.g an inertial navigation system) is to start with an initial position and then update it by cumulating measured movements. Terrain-aided positioning uses the terrain, more specifically the terrain height variation, to extract the aircraft position. Measured terrain height, provided by taking the difference between altitude over sea-level and ground clearance, is compared to the terrain height obtained from a database. The model of the system is

$$\begin{aligned} x_{t+1} &= x_t + u_t + v_t^x \\ z_{t+1} &= z_t + v_t^z \\ y_t &= h(x_t) + z_t + e_t, \end{aligned} \tag{1}$$

where x_t represent two-dimensional horizontal position and z_t represents altitude bias. The horizontal movement provided by the dead-reckoning system is denoted u_t. Note that (1) does not need a u_t^z because z_t represents not altitude, but altitude error. To be able to concentrate on position only the error inherent in measured movement, $v_t = (v_t^x, v_t^z)^T$, is considered to be white noise. In practice this is not true, e.g. the INS velocity error is highly correlated over time, see [4] how to deal with this. Moreover, in (1), y_t represents measured terrain elevation and $h(\cdot)$ is the terrain elevation database. The measurement noise, e_t, can be described by a Gaussian mixture with two modes; each mode corresponding to if the radar beam measuring the ground clearance hits the ground or the tree tops.

The aim is to recursively estimate the filtering posterior density $p(x_t, z_t|Y_t)$, where $Y_t = \{y_0, \dots, y_t\}$. Due to the not only nonlinear but also nonanalytical function $h(\cdot)$ the extended Kalman filter is practically out of the question. A feasible way to proceed is to apply the particle filter. As will be shown, applying it on the entire problem will lead to an unnecessarily high computational load. By considering (1) one sees that the altitude state enters the estimation problem linearly. This implies that it should be possible to estimate z_t using standard linear methods, and thereby reducing the dimension of the problem on which the particle filter is applied. Intuitively, this should in turn decrease the computational load caused by the normally rather computer intensive particle filter.

3. THE PARTICLE FILTER

Recursive Monte Carlo filtering methods, also referred to as particle filters, provide a solution to the general nonlinear, non-Gaussian filtering problem. Consider the state space model

$$\begin{aligned} x_{t+1} &= f(x_t) + v_t, \\ y_t &= h(x_t) + e_t, \end{aligned} \tag{2}$$

where the process and measurement noises, $v_t \sim p_{v_t}(\cdot)$ and $e_t \sim p_{e_t}(\cdot)$ respectively, are assumed independent with known but arbitrary densities. The state of the system, $X_t = \{x_0, \dots, x_t\}$, is a Markov process with $p(x_0|x_{-1}) = p(x_0)$, and the observations, $Y_t = \{y_0, \dots, y_t\}$, are conditionally independent given the states

$$p(X_t) = \prod_{k=0}^{t} p(x_k|x_{k-1}), \quad p(Y_t|X_t) = \prod_{k=0}^{t} p(y_k|x_k).$$

The aim is to recursively estimate the filtering density $p(x_t|Y_t)$, using the time and measurement recursions

$$\begin{aligned} p(x_{t+1}|Y_t) &= \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t \\ p(x_{t+1}|Y_{t+1}) &= \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|Y_t)}{p(y_{t+1}|Y_t)}, \end{aligned} \tag{3}$$

where $p(y_{t+1}|Y_t) = \int p(y_{t+1}|x_{t+1})p(x_{t+1}|Y_t)dx_{t+1}$.

In the general case there do not exist closed-form expressions for (3). Instead we are forced to use numerical methods. The principle for recursive Monte Carlo methods is to discretize the density, in a stochastic manner, utilizing a large number of samples. Suppose we have a set of independent samples $\{x_t^{(i)}\}_{i=1}^N$ with associated weights $\{w_t^{(i)}\}_{i=1}^N$, which together represent a Monte Carlo approximation of $p(x_t|Y_t)$, that is

$$p(x_t|Y_t) \approx \frac{\sum_{i=1}^N w_t^{(i)} \delta_{x_t^{(i)}}(x_t)}{\sum_{i=1}^N w_t^{(i)}} = \sum_{i=1}^N \bar{w}_t^{(i)} \delta_{x_t^{(i)}}(x_t), \tag{4}$$

where

$$\delta_{x_t^{(i)}}(x_t) = \begin{cases} 1 & \text{if } x_t = x_t^{(i)} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The set of samples with associated weights is said to be properly weighted with respect to $p(x_t|Y_t)$ if

$$\lim_{N\to\infty} \frac{\sum_{i=1}^N w_t^{(i)} g(x_t^{(i)})}{\sum_{i=1}^N w_t^{(i)}} = E_{p(x_t|Y_t)}(g(x_t)), \tag{6}$$

for any integrable function g. Plugging (4) into the recursion formulas (3) we obtain

$$p(x_{t+1}|Y_{t+1}) \approx \frac{p(y_{t+1}|x_{t+1})}{p(y_{t+1}|Y_t)} \sum_{i=1}^N \bar{w}_t^{(i)} p(x_{t+1}|x_t^{(i)}) \tag{7}$$

From (7), one way to recursively create a new set of independent, properly weighted samples is to draw from $p(x_{t+1}|x_t^{(i)})$, i.e.

$$x_{t+1}^{(i)} \sim p(x_{t+1}|x_t^{(i)}). \tag{8}$$

This gives that the associated weights are updated according to

$$w_{t+1}^{(i)} = p(y_{t+1}|x_{t+1}^{(i)})\bar{w}_t^{(i)}, \quad \bar{w}_{t+1}^{(i)} = \frac{w_{t+1}^{(i)}}{\sum_{i=1}^N w_{t+1}^{(i)}}. \tag{9}$$

In many cases there are more efficient ways to create new samples, see [2, 3, 5].

It can be shown [6] that the unconditional variance of the weights can only increase with time, meaning that most of the weights will tend to zero and leaving only a few that significantly represent the target density. To avoid this sample impoverishment we resample among the particles at regular intervals. Resampling can be done in many different ways, but one strategy known to have nice properties is residual resampling. The principle is to first multiply/discard particles according to $\lfloor N\bar{w}_t^{(i)} \rfloor$, and then to pick randomly, with replacement, amongst the rest, $M_t = N - \sum_{i=1}^N \lfloor N\bar{w}_t^{(i)} \rfloor$. In the last step, the probability for picking particle i is $M_t^{-1}(N\bar{w}_t^{(i)} - \lfloor N\bar{w}_t^{(i)} \rfloor)$. See [3, 7] for alternative resampling strategies.

A standard method for choosing when to resample is to use an approximative expression for the effective sample size of the filter [6]

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\bar{w}_t^{(i)})^2}. \tag{10}$$

When the effective sample size falls below a threshold, $\hat{N}_{\text{eff}} < N_{\text{h}}$ a resampling is performed.

The minimum mean square (MMS) estimate of x_t and its covariance are computed according to

$$\hat{x}_t^{\text{MMS}} \approx \sum_{i=1}^N \bar{w}_t^{(i)} x_t^{(i)} \tag{11}$$

$$P_t^{\text{MMS}} \approx \sum_{i=1}^N \bar{w}_t^{(i)} (x_t^{(i)} - \hat{x}_t^{\text{MMS}})(x_t^{(i)} - \hat{x}_t^{\text{MMS}})^T. \tag{12}$$

4. RAO-BLACKWELLIZATION

The purpose of Rao-Blackwellization is to try to reduce the number of particles needed for a given estimation precision.

The trick here is to apply the particle filter for estimation of horizontal position (x_t) only. Consider for the moment the stacked vector of position history, $X_t = \{x_0, \dots, x_t\}$. The posterior density for X_t and z_t can be factorized, using Bayes' rule, according to

$$p(X_t, z_t|Y_t) = p(z_t|X_t, Y_t)p(X_t|Y_t). \tag{13}$$

From (13) we see that if there exists a closed-form expression for $p(z_t|X_t, Y_t)$, and if there is a method for recursively updating $p(X_t|Y_t)$, we can reduce the dimension on which we apply the particle filter, and thereby we should be able to reduce the number of particles needed for a certain accuracy.

For the filtering density for altitude bias, we can use (4), where x_t now represents horizontal position, and extend it to cover the entire position history,

$$p(X_t|Y_t) = \sum_{i=1}^N \bar{w}_t^{(i)} \delta_{X_t^{(i)}}(X_t). \tag{14}$$

Plugging this into (13) gives

$$p(z_t|Y_t) = \sum_{i=1}^{N} \bar{w}_t^{(i)} p(z_t|X_t^{(i)}, Y_t). \tag{15}$$

How to estimate $p(z_t|X_t^{(i)}, Y_t)$ is described in Section 4.1, and how to recursively update the approximation of $p(X_t|Y_t)$ is described in Section 4.2.

4.1. Altitude estimation

Suppose here that the sequence $X_t = \{x_0, \dots, x_t\}$ is given. In practice, it is the sequence $X_t^{(i)}$ obtained from the particle filter that is given, but we leave the index out in the following for notational convenience. With $\xi_t = y_t - h(x_t)$ the part of (1) applicable for z_t is

$$\begin{aligned} z_{t+1} &= z_t + v_t^z \\ \xi_t &= z_t + e_t. \end{aligned} \tag{16}$$

Assume that $p_{v_t^z}(\cdot) = \mathcal{N}(v_t^z; 0, Q_t^z)$, where $\mathcal{N}(\cdot; m, Q)$ represents the Gaussian density with mean m and covariance Q. As mentioned in Section 2 the measurement noise e_t is not Gaussian, but rather a Gaussian mixture with two modes, i.e.

$$p_{e_t}(\cdot) = \sum_{k=0}^{1} P(\gamma_t = k) \mathcal{N}(e_t; m_t^k, R_t^k), \tag{17}$$

where γ_t is a binary Markov process with

$$P(\gamma_t = k) = p_k, \quad P(\gamma_t = k|\gamma_{t-1} = l) = \pi_{kl},$$

for $k, l = 0, 1$. Because of the exponential growth of possible mode sequences, $\Gamma_t = \{\gamma_0, \dots, \gamma_t\}$, the exact analytical solution for $p(z_t|\Xi_t)$, where $\Xi_t = \{\xi_0, \dots, \xi_t\}$, is intractable.

The simplest approach is to consider the generalized pseudo-Bayesian method of the first order (GPB1) [8]. The approximate solution is

$$p(z_t|\Xi_t) \approx \sum_{k=0}^{1} \mathcal{N}(z_t; \hat{z}_{t|t}^k, P_{t|t}^{z,k}) \bar{\alpha}_{t|t}^k, \tag{18}$$

where

$$\begin{aligned} P_{t|t-1}^z &= P_{t-1|t-1}^z + Q_{t-1}^z \\ S_t^k &= P_{t|t-1}^z + R_t^k \\ \hat{z}_{t|t}^k &= \hat{z}_{t-1|t-1} + P_{t|t-1}^z (S_t^k)^{-1} (\xi_t - m_t^k - \hat{z}_{t-1|t-1}) \\ P_{t|t}^{z,k} &= P_{t|t-1}^z - P_{t|t-1}^z (S_t^k)^{-1} P_{t|t-1}^z \end{aligned} \tag{19}$$

and

$$\begin{aligned} \alpha_{t|t}^k &= \overbrace{p(\xi_t|\gamma_t = k, \Xi_{t-1})}^{\approx \mathcal{N}(\xi_t; \hat{z}_{t-1|t-1} + m_t^k, S_t^k)} \sum_{l=0}^{1} \pi_{kl} \bar{\alpha}_{t-1|t-1}^l \\ \bar{\alpha}_{t|t}^k &= \frac{\alpha_{t|t}^k}{\sum_{k=0}^{1} \alpha_{t|t}^k}. \end{aligned} \tag{20}$$

Finally, the two Gaussian distributions are merged

$$\begin{aligned} \hat{z}_{t|t} &= \sum_{k=0}^{1} \bar{\alpha}_{t|t}^k \hat{z}_{t|t}^k \\ P_{t|t}^z &= \sum_{k=0}^{1} \bar{\alpha}_{t|t}^k (P_{t|t}^{z,k} + (\hat{z}_{t|t}^k - \hat{z}_{t|t})(\hat{z}_{t|t}^k - \hat{z}_{t|t})^T). \end{aligned} \tag{21}$$

4.2. Horizontal position estimation

Returning to (13), it remains to compute is $p(X_t|Y_t)$. This density can be rewritten recursively, using Bayes' rule repeatedly, according to

$$p(X_t|Y_t) = \frac{p(y_t|X_t, Y_{t-1}) p(x_t|x_{t-1})}{p(y_t|Y_{t-1})} p(X_{t-1}|Y_{t-1}). \tag{22}$$

In (22), the time propagation density is given by $p(x_t|x_{t-1}) = p_{v_t^x}(x_t - x_{t-1} - u_{t-1})$, not necessarily, but usually, assumed Gaussian distributed. Regarding $p(y_t|X_t, Y_{t-1})$, using the mode indicator γ_t we get

$$\begin{aligned} & p(y_t|X_t, Y_{t-1}) = \\ & \sum_{k=0}^{1} p(y_t|\gamma_t = k, X_t, Y_{t-1}) P(\gamma_t = k|X_t, Y_{t-1}) = \\ & \sum_{k=0}^{1} p(\xi_t|\gamma_t = k, \Xi_{t-1}) \sum_{l=0}^{1} \pi_{kl} P(\gamma_{t-1} = l|\Xi_{t-1}). \end{aligned} \tag{23}$$

Using the result from the GPB1 method given in (20) we obtain

$$p(y_t|X_t, Y_{t-1}) \approx \sum_{k=0}^{1} \alpha_{t|t}^k. \tag{24}$$

In other words, $p(y_t|X_t, Y_{t-1})$ is approximated by the sum of the unnormalized weights $\alpha_{t|t}^k$.

To summarize, we can estimate $p(X_t|Y_t)$ using the particle filter algorithm described in Section 3, only changing the weight update from (9) to

$$w_t^{(i)} = p(y_t|X_t^{(i)}, Y_{t-1}) \bar{w}_{t-1}^{(i)} = \sum_{k=0}^{1} \alpha_{t|t}^{k,(i)} \bar{w}_{t-1}^{(i)}. \tag{25}$$

5. SIMULATIONS

The whole idea of trying to solve as much of the problem as possible using closed-form methods is of course to reduce the computational load required for a given accuracy.

We performed a number of simulations to compare the two methods, i.e. the particle filter applied to 3-dimensional position and the particle filter applied to 2-dimensional horizontal position together with the GPB1 filter applied to altitude.

All the simulations were performed using one and the same trajectory considered to have adequate terrain variation. The simulation parameters are given in Table 1. To deal with density discretization errors, particularly during the convergence phase, the model of the process noise for horizontal position is extended to

$$v_t^{x,\text{model}} = v_t^{x,\text{system}} + v_t^{x,\text{add}}, \tag{26}$$

where $v_t^{x,\text{add}} \sim \mathcal{N}(0, k \cdot P_t^{\text{MMS},x})$ with $k = 0.001$ and $P_t^{\text{MMS},x}$ given by (12). To draw $\{x_0^{(i)}\}_{i=1}^N$ we use the true distribution, $p(x_0)$. To show that the particle filter combined with a GPB1 filter is robust with respect to $p(z_0)$ we chose $z_0^{(i)} = 0$, $P_0^{z,(i)} = 100^2$ for all $i = 1, \dots, N$. The measurement update rate of the filters are in all cases 1 Hz.

The result from the simulations is shown in Table 2, where the result is based on 100 Monte Carlo simulations for each filtering method and number of particles. The simulations were performed

Table 1. Simulation parameters

u_t	$\begin{bmatrix}-100 & 100 & 0\end{bmatrix}^T$ m/s
$p_{e_t}(\cdot)$	$0.5 \cdot \mathcal{N}(e_t; 0, 3^2) + 0.5 \cdot \mathcal{N}(e_t; 12, 6^2)$
$p_{v_t^x}(\cdot), p_{v_t^z}(\cdot)$	$\mathcal{N}\left(v_t^x; \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2^2 & 0\\0 & 2^2\end{bmatrix}\right), \mathcal{N}(v_t^z; 0, 0.2^2)$
$p(x_0)$	$\begin{cases} 1/2000^2 & \text{if } \lvert x_0^1\rvert < 1000, \lvert x_0^2\rvert < 1000 \\ 0 & \text{otherwise}\end{cases}$
$p(z_0)$	$\begin{cases} 1/(200\sqrt{3}) & \text{if } \lvert z_0\rvert < 100 \cdot \sqrt{3} \\ 0 & \text{otherwise}\end{cases}$
π_{kl}	0.5 for $k, l = 0, 1$

in Matlab on a Sun station (Ultra 10). The table shows that by using the particle and GPB1 filter it is enough to use less than half the number of particles compared to the stand-alone particle filter for the same accuracy (24 m for horizontal position and 1.0 m for altitude). Less than half the number of particles in this case means that the computational load is about 65% of the load for the stand-alone particle filter. These results are also confirmed by Figure 1,

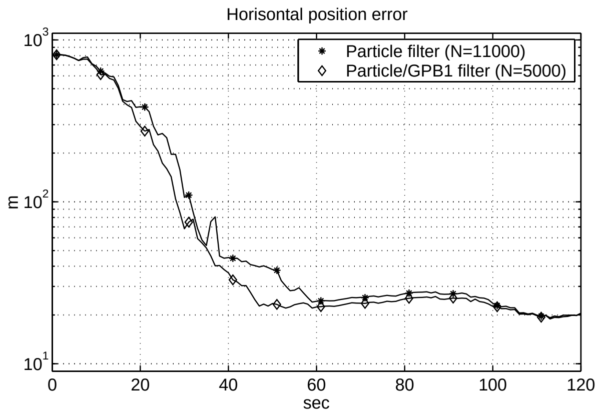
Table 2. Required simulation time and $\sqrt{(\frac{1}{61}\sum_{t=60}^{120} \mathrm{RMSE}(\cdot)^2)}$ for $\hat{x}_t^{\mathrm{MMS}}$ and $\hat{z}_t^{\mathrm{MMS}}$ as a function of method and number of particles.

	N	**Time**	**Position**	**Altitude**
Particle filter	10000	24.6 s	76.2 m	0.99 m
	11000	26.6 s	24.5 m	0.97 m
	12000	29.7 s	24.2 m	0.99 m
Particle / GPB1 filter	4000	14.3 s	80.8 m	1.0 m
	5000	17.7 s	23.1 m	0.96 m
	6000	20.9 s	23.0 m	0.96 m

which shows typical behaviour of the RMSE of horizontal position and altitude over time for the two filters.

6. CONCLUSIONS

In this paper we have given two ways to estimate three-dimensional position given a dead-reckoning system and a terrain-aided positioning system. Both methods are based on the particle filter, what differs is the way we estimate altitude. We have shown that the particle filter is well suited for this highly nonlinear problem, but the computational load is unnecessarily high. By using the fact that altitude enters linearly we can use Rao-Blackwellisation technique to reduce the dimension on which we apply the particle filter, and thereby reduce the computational load.



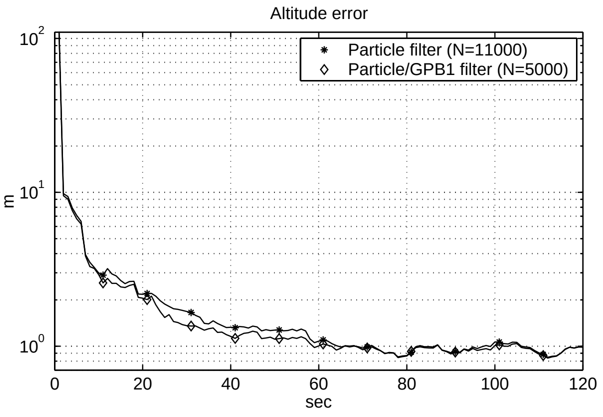


Fig. 1. RMSE of $\hat{x}_t^{\mathrm{MMS}}$ (upper) and $\hat{z}_t^{\mathrm{MMS}}$ (lower) based on 100 Monte Carlo simulations for the stand-alone particle filter (11000 particles) compared with the Rao-Blackwellized particle filter (5000 particles) using GPB1 technique to estimate altitude.

7. REFERENCES

[1] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings on Radar and Signal Processing*, 1993, vol. 140, pp. 107–113.

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

[3] J.S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statisticial Association*, vol. 93, no. 443, September 1998.

[4] P-J. Nordlund and F. Gustafsson, "Sequential Monte Carlo filtering techniques applied to integrated navigation systems," in *Proceedings of the 2001 American Control Conference*, 2001, vol. 6, pp. 4375–4380.

[5] M.K. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, pp. 590–599, 1999.

[6] A. Kong, J.S. Liu, and W.H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 524, pp. 278–288, March 1994.

[7] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[8] Y. Bar-Shalom and X-R. Li, *Estimation and Tracking: Principles, Techniques and Software,* Artech House, Boston, London, 1993.