Deterministic and Stochastic Bayesian Methods in Terrain Navigation

Niclas Bergman

Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden
WWW: <http://www.control.isy.liu.se>
Email: niclas@isy.liu.se

August 28, 1998

Report no.: LiTH-ISY-R-2052

Submitted to IEEE Conf. on Decision and Control, Tampa 1998

Technical reports from the Automatic Control group in Linköping are available by anonymous ftp at the address <ftp.control.isy.liu.se>. This report is contained in the compressed postscript file 2052.ps.Z.

Deterministic and Stochastic Bayesian Methods in Terrain Navigation

Niclas Bergman
Dept. of Electrical Engineering
Linköping University
SE 58183 Linköping, Sweden
E-mail: niclas@isy.liu.se

Abstract

Terrain navigation is an application where inference between conceptually different sensors is performed recursively on-line. In this work the Bayesian framework of statistical inference is applied to this recursive estimation problem. Three algorithms for approximative Bayesian estimation are evaluated in simulations, one deterministic algorithm and two stochastic. The deterministic method solve the Bayesian inference problem by numerical integration while the stochastic methods simulate several candidate solutions and evaluates the integral by averaging between these candidates. Simulations show that all three algorithms are efficient and approximately reach the Cramér-Rao bound. However, the stochastic methods are sensitive to outliers and the deterministic method has the limitation of being hard to implement in higher dimensions.

1 Introduction

The Bayesian methods have had a renaissance in signal processing due to many new and rediscovered algorithms that promises approximative but tractable solutions to high dimensional estimation problems [5]. The increase of computer processing power has also made older algorithms feasible for implementation. In this work we investigate the performance of some algorithms for Bayesian inference applied to terrain navigation.

Terrain navigation is a technique for autonomous aircraft navigation used extensively as the main or complementary navigation system in military crafts and missiles. Measurements from different navigation sensors are fused together and compared with a digital reference terrain map in order to generate a position estimate. The fusion process can be seen as a statistical inference problem and a sequential nonlinear estimation problem must be solved on-line in real time.

In the next section Bayesian inference based on sequen-

tial information processing for on-line estimation is reviewed. In Section 3 the basic principles of terrain navigation are explained and the conceptual Bayesian solution given. Section 4 lists three algorithms for Bayesian terrain navigation which are evaluated in simulations displayed in Section 5. The paper is concluded by some remarks on the experience gained during the project.

2 Sequential Bayesian Estimation

Consider a hidden Markovian process $\{x_t\}_{t \in \mathbb{N}}$ where $x_t \in \mathbb{R}^n$ has known transition kernel $p(x_{t+1} | x_t)$. We are interested in estimating this process based on the observations $\{y_t\}_{t \in \mathbb{N}}$ where $y_t \in \mathbb{R}^p$ is conditionally white given the state x_t and has known conditional distribution $p(y_t | x_t)$.

Let capital letters denote the complete history of the processes $X_t = \{x_i\}_{i=0}^t$ and $Y_t = \{y_i\}_{i=0}^t$. In a Bayesian context all relevant information about the process $\{x_t\}$ at time t is condensed in the posterior distribution $p(X_t | Y_t)$. In sequential Bayesian estimation we are interested in determining recursively in time any of its marginals, usually the filter density $p(x_t | Y_t)$.

Assume that $p(x_t | Y_{t-1})$ is known, using Bayes rule and recursively conditioning only on the last element in the set Y_t we get

$$p(x_t | Y_t) = \frac{p(y_t | x_t)p(x_t | Y_{t-1})}{p(y_t | Y_{t-1})} \quad (1)$$

where the denominator is a normalization factor

$$p(y_t | Y_{t-1}) = \int p(y_t | x_t)p(x_t | Y_{t-1}) dx_t.$$

Using the Markovian property of the process $\{x_t\}$

$$p(x_{t+1} | Y_t) = \int p(x_{t+1} | x_t)p(x_t | Y_t) dx_t, \quad (2)$$

the expressions (1) and (2) define a recursive formula for the Bayesian solution to the filtering problem given some prior distribution $p(x_0 | Y_{-1}) = p(x_0)$.

The filter density completely describes the characteristics of the states given the collected observations. For practical reasons a point estimate of the state is desirable. Introduce some cost function $L(x_t^*, x_t)$ that penalizes any given state candidate x_t^* different from the true state x_t . A Bayesian estimator minimizes the posterior expected cost given the collected measurements,

$$\hat{x}_t = \arg \min_{x_t^*} \int L(x_t^*, x_t) p(x_t | Y_t) dx_t.$$

Two common choices of penalty function $L(\cdot, \cdot)$ result in explicit expressions for the Bayesian estimate.

MMSE The minimum mean square error estimate where $L(x_t^*, x_t) = (x_t - x_t^*)^T Q (x_t - x_t^*)$ for any positive definite matrix Q yields

$$\hat{x}_t^{\text{MMSE}} = \int x_t p(x_t | Y_t) dx_t.$$

For obvious reasons, this estimate is also labeled the conditional mean.

MAP The maximum a posteriori estimate with the $1/\delta$ penalty function $L(x_t^*, x_t) = \mathbf{I}_{\{\|x_t^* - x_t\| > \delta\}}$ where as $\delta \rightarrow 0_+$ we obtain

$$\hat{x}_t^{\text{MAP}} = \arg \max_{x_t} p(x_t | Y_t).$$

In the simulations presented in Section 5 the MMSE estimate is used.

In spite of the appealing unifying framework of Bayesian estimation the number of applications in statistics, data analysis and statistical signal processing is limited. This stems from the fact that Bayesian inference can only be performed analytically in very rare cases. As seen above Bayesian estimation involves integration and/or optimization over often very complex and multidimensional functions. Only when these integrals and/or optimizations have explicit analytical solutions can the Bayesian inference be implemented exactly. Except for the linear Gaussian case almost no explicit solutions exist.

Thus, in almost all cases of nonlinear Bayesian estimation approximations are inevitable and all practical algorithms seek to solve the Bayesian inference with as little error as possible. The aim of the practical algorithm is twofold, first to approximate as closely as possible the posterior filter density $p(x_t | Y_t)$ and second to update this approximation with time and with each new measurement y_t . It is the latter requirement that makes sequential estimation somewhat more complicated than regular inference. The algorithms can usually be put in either of the two classes stochastic, or simulation based methods, and deterministic methods based on numerical integration.

2.1 Simulation Based Methods

The key idea in simulation based methods is to generate N i.i.d. candidate state vectors $\{x_t^i\}_{i=1}^N$ according to the posterior distribution $p(x_t | Y_t)$. If this can be done recursively in time then any integral

$$I(f) \triangleq \mathbf{E}(f(x_t) | Y_t) = \int f(x_t) p(x_t | Y_t) dx_t$$

can be estimated by the sum

$$I_N = \frac{1}{N} \sum_{i=1}^N f(x_t^i).$$

Applying the strong law of large numbers this estimate will converge $I_N \rightarrow I(f)$ almost surely as $N \rightarrow \infty$. Moreover, assuming that

$$\frac{\sigma_f^2}{N} \triangleq \frac{1}{N} \mathbf{E}((f(x_t) - \mathbf{E}f(x_t))^2 | Y_t) < \infty$$

the central limit theorem gives convergence in distribution to an asymptotically normal approximation error

$$\sqrt{N}(I_N - I(f)) \rightarrow \mathcal{N}(0, \sigma_f^2) \quad \text{as } N \rightarrow \infty.$$

See [6] for formal assumptions and proofs of these convergence statements. Note that no explicit dimension dependence appears in the convergence expressions so that N does not necessarily have to increase drastically with the dimension of the space. However, the number N might still have to be very large in practical cases.

If the MAP estimate is considered, the maximization can be performed among the N candidate vectors $\{x_t^i\}_{i=1}^N$. The rationale behind this being that the candidate vectors are drawn from $p(x_t | Y_t)$ and thus should be concentrated in areas of high probability around the maximum of the filter density.

Comprehensive reviews over simulation based methods in Bayesian statistics can be found in [5, 9]. Two algorithms using the recursive simulation method are applied to terrain navigation in Section 4.

2.2 Numerical Integration

The numerical integration procedures directly attacks the problem of evaluating the integrals in the Bayesian solution using quadrature formulas

$$\int f(x_t) p(x_t | Y_t) dx_t \approx \sum_{i=1}^N \alpha_i f(x_t^i) p(x_t^i | Y_t).$$

Usually the nodes x_t^i are chosen in a uniform mesh over the state space using the quadrature weights $\alpha_i = \delta^n$ where δ is the mesh resolution. Naturally, as $\delta \rightarrow 0$ the approximation converges to the true value of the integral assuming that $f(\cdot)$ is reasonably smooth such that the integral exist finitely. A simple analysis [4]

shows that using a uniform mesh the relative error is of order $O(N^{-1/n})$, thus in order to keep a fixed relative accuracy the number of grid points grows exponentially with the dimension of the state space. This fact is known as the *curse of dimensionality* and has a profound effect on the ability to apply numerical integration in higher dimensional spaces. However, in low dimensional spaces or with an adaptive choice of grid node positions this exponential growth of N can be attenuated.

Early work on numerical integration methods for recursive estimation was performed by Bucy and Senne [3] and later extended by Kramer and Sorenson [8]. Section 4 presents a numerical integration method with adaptive grid applied to terrain navigation.

3 Terrain Navigation

Navigation systems in modern aircraft are subject to high demands on reliability and accuracy. Therefore, modern navigation systems determine the position of the aircraft using information from several, partly redundant, sources. Such integrated navigation systems are usually configured around an Inertial Navigation System (INS) which is supported by several complementary sensors. An INS determines the aircraft position relative to an initial position by continuously measuring the aircraft movement, using accelerometers and gyroscopes. Due to the dead-reckoning configuration of the INS, measurement errors cannot be attenuated. Instead, the error in the position estimate from an INS tend to increase linearly with time.

Terrain navigation is a technique that, like an INS, autonomously determines the aircraft position. Terrain navigation is an example of an application where non-standard fusion of sensor data from several different information sources is needed. Consider Figure 1, the aircraft altitude over mean sea-level is determined by comparing the measurements from a pressure meter with the normal pressure at mean sea-level. The distance to the ground is measured using a radar altimeter. The terrain height over mean sea-level, i.e., the terrain elevation, is found from the difference between the altitude and the ground clearance. Since the terrain elevation is a function of longitude and latitude a digitalized map of the terrain can be used to find positions matching the measured terrain elevation thereby producing autonomously derived aircraft positions. In order to eliminate the false position matches several measurements must be collected and matched with the terrain database using information about the aircraft movement between consecutive measurements obtained from the INS. Since safety and reliability are two important issues in this application, high demands are put

Figure 1: The principle of terrain navigation is to measure the terrain elevation beneath the aircraft and compare this measurement with a digital terrain map.

on the sensor fusion filter that decides upon an aircraft position based on information from the pressure sensor, the radar altimeter, the INS and the digital map. By modeling the problem in a statistical framework and applying the Bayesian solution a complete description of the aircraft position probability distribution given the collected measurements is obtained. The shape of this function gives a lot of information about any point estimate delivered by the system, thereby increasing the reliability of the total system.

Let the vector $x_t \in \mathbb{R}^2$ denote the position of the aircraft in the map, and y_t denote the scalar measured terrain elevation. The state evolution in time and its relation to the measurements are described by the model

$$\begin{aligned} x_{t+1} &= x_t + u_t + v_t \\ y_t &= h(x_t) + e_t \end{aligned} \quad t = 0, 1, \dots \quad (3)$$

where $h(\cdot)$ is the terrain map, u_t is the movement of the aircraft estimated by the INS, v_t is the error drift in the INS, and e_t is the measurement noise and error in the terrain map. The processes $\{v_t\}$ and $\{e_t\}$ are assumed white and independent of each other, they are also assumed uncorrelated with the state at time $t = 0$. The Bayesian recursive solution to terrain navigation is found by inserting (3) into (1) and (2) yielding

$$\begin{aligned} p(x_t | Y_t) &= c_t p_{e_t}(y_t - h(x_t)) p(x_t | Y_{t-1}) \\ p(x_{t+1} | Y_t) &= \int p_{v_t}(x_{t+1} - x_t - u_t) p(x_t | Y_t) dx_t \end{aligned} \quad (4)$$

where c_t is a normalization constant. The recursion is initiated by the density function of the states at time $t = 0$, $p(x_0 | y_{-1}) = p(x_0)$.

4 Algorithms

This section summarizes three algorithms for sequential Bayesian estimation applied to the terrain navigation problem.

4.1 The Bootstrap filter

The Bootstrap filter [7] is a simulation based method for Bayesian estimation where a large set of random samples of the state vector are used to represent the conditional density function. On the reception of a new measurement resampling with replacement is performed from this set where each sample is drawn with a probability determined by the likelihood of the measurement.

Algorithm 1 (Bootstrap filter)

Initialize *Simulate* $x_0^{i*} \sim p(x_0)$ for $i = 1, \dots, N$ and let $t = 0$.

Update *Calculate the normalized weights*

$$w_i = \frac{p_{e_t}(y_t - h(x_t^{i*}))}{\sum_{j=1}^N p_{e_t}(y_t - h(x_t^{j*}))} \quad i = 1, \dots, N$$

and determine the estimate $\hat{x}_t = \sum_{i=1}^N w_i x_t^{i*}$.

Resample *Let* $\{w_i\}$ *define a discrete distribution over* $\{x_t^{i*}\}_{i=1}^N$ *and resample with replacement* N *times from this set to generate the new set of samples* $\{x_t^j\}_{j=1}^N$ *where* $\Pr(x_t^j = x_t^{i*}) = w_i$ *for any* j .

Predict *Generate* $v_t^i \sim p(v_t)$ *for* $i = 1, \dots, N$ *and compute*

$$x_{t+1}^{i*} = x_t^i + u_t + v_t^i \quad i = 1, \dots, N.$$

Set $t := t + 1$ and continue at the update step.

The simplicity of the algorithm makes it fast and easy to implement, the main computational burden lies in the resampling step. A more advanced simulation algorithm is described in the section below where the resampling is performed only when need.

4.2 Importance Sampling

Ideally one would like to generate N samples $\{x_t^i\}_{i=1}^N$ i.i.d. according to the filter density recursively in time. Often this cannot be done but instead one can generate the samples according to some other distribution $\pi(x_t | Y_t)$ labeled the *importance function*. Then the sought integral can be written

$$I(f) = \int f(x_t) \frac{p(x_t | Y_t)}{\pi(x_t | Y_t)} \pi(x_t | Y_t) dx_t$$

and evaluated using the weighted sum

$$I_N = \sum_{i=1}^N w_t^i f(x_t^i), \quad w_t^i = \frac{p(x_t^i | Y_t)}{\pi(x_t^i | Y_t)}.$$

In Bayesian recursive filtering one works with unnormalized densities and the weights are updated recursively in time. Due to the normalization and the recursive update the weights will in general degenerate so that only a few weights contributes to the sum while the others are close to zero [5]. Therefore a Bootstrap resampling is introduced whenever the estimated effective number of samples

$$\hat{N}_{\text{eff}} \triangleq \frac{1}{\sum_{i=1}^N w_t^i{}^2}$$

falls below some threshold N_{thresh} [5]. The importance function is arbitrary as long as it covers the support of the filter density, in this work we use the prior as importance function in the recursive update. There exist several enhanced versions of sequential importance sampling algorithms often these seek to evaluate the optimal importance function, see [5, 9] for a complete list. The algorithm for Bayesian prior importance sampling applied to the terrain navigation problem is outlined below.

Algorithm 2 (Sequential Importance Sampling)

Initialize *Sample* $x_0^i \sim p(x_0)$ and set $w_{-1}^i = \frac{1}{N}$ for $i = 1, \dots, N$. Let $t = 0$.

Update *Compute the weights*

$$w_t^i := w_{t-1}^i p_{e_t}(y_t - h(x_t^i)) \quad i = 1, \dots, N$$

and normalize the result

$$w_t^i := \frac{w_t^i}{\sum_{i=1}^N w_t^i} \quad i = 1, \dots, N.$$

Determine the estimate $\hat{x}_t = \sum_{i=1}^N w_t^i x_t^i$.

Resample *If* $\hat{N}_{\text{eff}} \leq N_{\text{thresh}}$ *resample with replacement from the set* $\{x_t^i\}_{i=1}^N$ *where* w_t^i *is the probability to resample node* i .

Predict *Generate* $v_t^i \sim p(v_t)$ *for* $i = 1, \dots, N$ *and compute*

$$x_{t+1}^i = x_t^i + u_t + v_t^i \quad i = 1, \dots, N.$$

Set $t := t + 1$ and repeat at the update step above.

Compared to the Bootstrap filter, the weights in sequential importance sampling are recursively determined and resampling is only performed when actually needed.

4.3 The point-mass filter

While simulation based methods utilize the model (3) generating a large number of candidate tracks and calculating some probability weight for each track, numerical integration directly attacks the solution (4) seeking

to evaluate the integral numerically and update a description of the filter density recursively in time. The numerical integration method used in this work uses a simple quadrature formula with constant weights for evaluating the recursive solution (4) in a uniform grid.

Let $\{x_t^i\}_{i=1}^N$ be points chosen from a uniform mesh with resolution δ , i.e., $x_t^i \in \{\delta k : k \in \mathbb{Z}^2\}$. Each point has a corresponding probability mass denoted p_t^i . The measurement update, normalization and estimate calculation are straightforwardly implemented. Due to the linear state transition equation in (3) the integral in (4) is a linear convolution. By discretizing the kernel $p(v_t)$ using the mesh resolution δ the density can be updated by standard discrete convolution. The number of grid points will thereby be increased around the borders of the support of $\{x_t^i\}_{i=1}^N$. To compensate for this the mesh is adapted after each measurement update by removing all grid points that fall below the average grid point value by some fraction ε . For more details on point-mass filters using adaptive mesh resolution see [1, 2].

Algorithm 3 (Point-mass filter)

Initialize Evaluate $p_0^i = p(x_0^i)$ for $i = 1, \dots, N_0$ and set $t = 0$.

Update Calculate

$$p_t^i := p_t^i p_{e_t}(y_t - h(x_t^i)) \quad i = 1, \dots, N_t$$

and normalize $p_t^i := p_t^i / \sum_{j=1}^{N_t} p_t^j \delta^2$. Calculate the estimate $\hat{x}_t = \sum_{i=1}^{N_t} p_t^i x_t^i$.

Refine Remove all points where $p_t^i < \varepsilon / N_t \delta^2$ and calculate the number of remaining grid points N_t .

Predict Evaluate the two dimensional discrete convolution

$$p_{t+1}^i = \sum_{j=1}^{N_t} p_{v_t}(x_{t+1}^i - x_t^j - u_t) p_t^j \delta^2$$

using the set $\{x_{t+1}^i\}_{i=1}^{N_{t+1}}$ chosen over the support of the convolution result. Set $t := t+1$ and repeat at the update step above.

The main difference between the point-mass filter and the two previous simulation based methods is that the grid is chosen by the user. This has the disadvantage of more complicated implementation but the advantage of less sensitivity to outliers. The algorithm is rather simple to implement in one or two dimensions but in higher dimensions the implementation complexity becomes severe.

5 Simulation Evaluation

A commercial terrain map was used to evaluate the performance of the algorithms. The map covers a part of central Sweden and is stored in a uniform grid of 50 m spacing.

The algorithms are evaluated in Monte Carlo simulations, the aircraft tracks were generated using Gaussian distributed initial aircraft position and zero mean Gaussian noises $\{v_t\}$ and $\{e_t\}$, the parameters are listed in Table 1. The simulation tracks are displayed

Initial covariance	$P_0 = 100^2 I_2 \text{ m}^2$
State noise covariance	$Q = 5^2 I_2 \text{ m}^2$
Measurement noise covariance	$R = 16 \text{ m}^2$
Track length	150 samples
Aircraft velocity	$[25, 25]^T \text{ m/sample}$
Monte Carlo runs	100

Table 1: Simulation track parameters.

in Figure 2 over a contour plot of the commercial map.

Figure 2: The 100 simulation tracks over a contour description of the commercial map.

The importance sampling algorithm used $N_{\text{thresh}} = N/2$ and the point-mass filter $\varepsilon = 10^{-3}$ and $\delta = 12.5 \text{ m}$. The simulation based algorithms are very sensitive to outliers, the frequency of lost tracks during 100 Monte Carlo simulations are listed in Table 2. The numerical integration point-mass filter experienced no lost tracks during the simulations. As mentioned above safety and reliability are important issues in navigation, thus the point-mass filter is to be preferred on this

N	Bootstrap	Import. Sampl.
50	3	3
100	1	4
200	1	3
400	0	1

Table 2: Number of lost tracks during Monte Carlo simulations.

point. The bootstrap filter is more robust than the importance sampling method but the bootstrap filter demands much more computations. In the importance sampling algorithm resampling is only performed during some 20% of the iterations and hence the computational load is approximately one fifth of that of the bootstrap filter.

The estimation accuracy increases gradually with increasing number of simulation candidates N in the stochastic methods. However, above $N = 200$ the difference is negligible. The Root Mean Square (RMS) error of the algorithms are compared in Figure 3 where

Figure 3: Monte Carlo RMS error and the posterior Cramér-Rao bound.

the lost tracks have been removed. Both simulation based methods used $N = 400$ samples in the plots shown in Figure 3, the point-mass filter had on the average 339 grid points during these simulations. For comparative reasons the plot also show the posterior Cramér-Rao bound [10] which is a fundamental lower limit on the RMS error. As seen in the figure, all three algorithms approximately reach the lower bound after an initial convergence phase. Hence, all three approximative algorithms solve the Bayesian inference problem with best possible performance.

6 Conclusions

Bayesian inference is a general framework for fusion of sensor measurements that only rarely can be solved analytically. The approximative algorithms evaluated in this work show that the Bayesian approach can, with success, be applied to nonlinear recursive estimation. The numerical integration method shows greater insensitivity to outliers but is very hard to implement in higher dimensions, e.g., if one would like to estimate the aircraft velocity along with its position. The simulation based methods are simple to implement even in very high dimension but has the disadvantage of losing the true aircraft trajectory occasionally. All three algorithms shows effectiveness reaching the Cramér-Rao lower bound.

References

- [1] N. Bergman. *Bayesian Inference in Terrain Navigation*. Linköping Studies in Science and Technology. Thesis No 649, 1997.
- [2] N. Bergman, L. Ljung, and F. Gustafsson. Point-mass filter and Cramer-Rao bound for terrain-aided navigation. In *Proc. 36:th IEEE Conf. on decision and control*, pages 565–570, 1997.
- [3] R.S. Bucy and K.D. Senne. Digital synthesis of nonlinear filters. *Automatica*, 7:287–298, 1971.
- [4] P.J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, 2nd edition, 1984.
- [5] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report TR.310, Signal Processing Group, Department of Engineering, University of Cambridge, 1998.
- [6] J. Geweke. Bayesian inference in econometrics models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339, 1989.
- [7] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.
- [8] S.C. Kramer and H.W. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24:789–801, 1988.
- [9] J.S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93, 1998.
- [10] P. Tichavský, C. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.