ONLINE EXPECTATION-MAXIMIZATION TYPE ALGORITHMS FOR PARAMETER ESTIMATION IN GENERAL STATE SPACE MODELS

Christophe Andrieu† - Arnaud Doucet‡
†Department of Mathematics, Bristol University,
Bristol, BS8 1TW, UK.
‡Department of Engineering, Cambridge University
Cambridge, CB2 1PZ, UK.
Email: C.Andrieu@bristol.ac.uk - ad2@eng.cam.ac.uk

ABSTRACT

In this paper we present new online algorithms to estimate static parameters in nonlinear non Gaussian state space models. These algorithms rely on online Expectation-Maximization (EM) type algorithms. Contrary to standard Sequential Monte Carlo (SMC) methods recently proposed in the literature, these algorithms do not degenerate over time.

1 Introduction

1.1 State-Space Models and Problem Statement

Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 1}$ be $\mathbb{R}^p$ and $\mathbb{R}^q$-valued stochastic processes defined on a measurable space $(\Omega, \mathcal{F})$ and $\theta \in \Theta$ where Θ is an open subset of $\mathbb{R}^k$. The process $\{X_n\}_{n\geq 0}$ is an unobserved (hidden) Markov process of initial density μ; i.e. $X_0 \sim \mu$, and Markov transition density $f_\theta(x, x')$; i.e.

$$X_{n+1} | X_n = x \sim f_\theta(x, \cdot). \quad (1)$$

One observes the process $\{Y_n\}_{n\geq 1}$. It is assumed that the observations are conditionally independent upon $\{X_n\}_{n\geq 0}$ of marginal density $g_\theta(x, y)$; i.e.

$$Y_n | X_n = x \sim g_\theta(x, \cdot). \quad (2)$$

This class of models includes many nonlinear and non-Gaussian time series models such as

$$X_{n+1} = \varphi_\theta(X_n, V_{n+1}),\ Y_n = \psi_\theta(X_n, W_n)$$

where $\{V_n\}_{n\geq 1}$ and $\{W_n\}_{n\geq 1}$ are independent sequences.

We assume here that the model depends on some unknown parameter denoted θ. The true value of θ is θ^*. We are interested in deriving recursive algorithms to estimate θ^* for the class of stationary state-space models; i.e. when the Markov chain $\{X_n\}_{n\geq 0}$ admits a limiting distribution. This problem has numerous applications in electrical engineering, econometrics, statistics etc. It is extremely complex. Even if θ^* was known, the simpler problem of optimal filtering, i.e. estimating the posterior distribution of X_n given $(Y_1, \ldots, Y_n)$, does not usually admit a closed-form solution.

Further on we will denote for any sequence z_k/random process Z_k $z_{i:j} = (z_i, z_{i+1}, \ldots, z_j)$ and $Z_{i:j} = (Z_i, Z_{i+1}, \ldots, Z_j)$.

1.2 A Brief Literature Review

1.2.1 Filtering methods

A standard approach followed in the literature consists of setting a prior distribution on the unknown parameter θ and then considering the extended state $S_n \triangleq (X_n, \theta)$. This converts the parameter estimation into an optimal filtering problem. One can then apply, at least theoretically, standard particle filtering techniques [5] to estimate $p(x_n, \theta | Y_{1:n})$ and thus $p(\theta | Y_{1:n})$. In this approach, the parameter space is only explored at the initialization of the algorithm. Consequently the algorithm is inefficient; after a few iterations the marginal posterior distribution of the parameter is approximated by a single delta Dirac function. To limit this problem, several authors have proposed to use kernel density estimation methods [13]. However, this has the effect of transforming the fixed parameter into a slowly time-varying one. A pragmatic approach consists of introducing explicitly an artificial dynamic on the parameter of interest; see [10], [11]. To avoid the introduction of an artificial dynamic model, an approach proposed in [8] consists of adding Markov chain Monte Carlo (MCMC) steps so as to add "diversity" among the particles. However, this approach does not solve the fixed-parameter estimation problem. More precisely, the addition of MCMC steps does not make the dynamic model ergodic. Thus, there is an accumulation of errors over time and the algorithm can diverge as observed in [1]. Similar problems arise with the recent method proposed in [16].

1.2.2 Recursive maximum likelihood

Consider the log-likelihood function

$$l_\theta(Y_{1:n}) = \sum_{k=1}^{n} \log\left(\int g_\theta(x_k, Y_k)\, p_\theta(x_k | Y_{1:k-1})\, dx_k\right). \quad (3)$$

where $p_\theta(x_k | Y_{1:k-1})$ is the posterior density of the state X_k given the observations $Y_{1:k-1}$ and θ. Under regularity assumptions including the stationarity of the state-space model, one has

$$\frac{1}{n} l_\theta(Y_{1:n}) \to l(\theta) \quad (4)$$

where

$$l(\theta) \triangleq \int\int_{\mathbb{R}^q \times \mathcal{P}(\mathbb{R}^p)} \log\left(\int g_\theta(x, y)\, \mu(x)\, dx\right) \lambda_{\theta,\theta^*}(dy, d\mu),$$

where $\mathcal{P}(\mathbb{R}^p)$ is the space of probability distributions on $\mathbb{R}^p$ and $\lambda_{\theta,\theta^*}(dy, d\mu)$ is the joint invariant distribution of the couple $(Y_k, p_\theta(x_k|Y_{1:k-1}))$. It is dependent on both θ and the true parameter θ^*. Maximizing $l(\theta)$ corresponds to minimizing the following Kullback–Leibler information measure given by

$$K(\theta, \theta^*) \triangleq l(\theta^*) - l(\theta) \geq 0. \quad (5)$$

To optimize this cost function, Recursive Maximum Likelihood (RML) is based on a stochastic gradient algorithm

$$\theta_{n+1} = \theta_n + \gamma_n \nabla \log \left(\int \varepsilon_{\theta_n}(x_n, Y_n) \, p_{\theta_{1:n}}(x_n|Y_{1:n-1}) \, dx_n \right).$$

This requires the computation of $p_{\theta_{1:n}}(x_n|Y_{1:n-1})$ and its derivatives with respect to θ using the parameter θ_k at time k where $p_{\theta_{1:n}}(x_n|Y_{1:n-1})$ denotes the predictive distribution computed using parameter θ_k at time k. This is the approach followed in [12] for finite state-space HMM and in [6] for general state-space models. In the general state-space case, the filter and its derivatives are approximated numerically. This method does not suffer from the problems mentioned above. However it is sensitive to initialization and in practice it can be difficult to scale the components of the gradient properly.

1.2.3 Online EM algorithm

Another stochastic gradient type algorithm is based on an online version of the EM. This approach is better from a practical point of view as it is easy to implement and numerically well-behaved. One can actually show that online EM corresponds to a Newton-Raphson like algorithm; the difference being that the gradient is not scaled by the information matrix but by the complete information matrix. Online EM algorithms have been proposed for finite state-space HMM and linear Gaussian state-space models; e.g. [7].

It is formally possible to come up with a similar algorithm for general state-space models. However, it requires one to compute quantities such as

$$\begin{aligned}
&\mathcal{Q}_n(\theta|\theta') = \mathbb{E}_{\theta'}(\log p_\theta(x_{0:n}, Y_{1:n})|Y_{1:n}) \\
&= \mathbb{E}_{\theta'}(\log \mu(x_0)|Y_{1:n}) + \sum_{k=1}^{n} \mathbb{E}_{\theta'}(\log f_\theta(x_{k-1}, x_k)|Y_{1:n}) \\
&+ \sum_{k=1}^{n} \mathbb{E}_{\theta'}(\log g_\theta(x_k, Y_k)|Y_{1:n})
\end{aligned}$$

It is trivially possible to estimate $\mathcal{Q}_n(\theta|\theta')$ using particle methods as one gets an estimate of the joint posterior density $p_\theta(x_{1:n}|Y_{1:n})$. However, as n increases, only the approximation of marginal densities $p_\theta(x_{n-L:n}|Y_{1:n})$ for a reasonable value of L, say L around 5, is good for a reasonable number of particles [5]. It seems impossible to obtain a good approximation of $\mathcal{Q}_n(\theta|\theta')$ with an online algorithm as n increases. It is thus necessary to come up with an alternative method.

1.3 Contributions

We propose here three new algorithms to address the problem of recursive parameter estimation in general state-space models. These algorithms rely on online EM type algorithms, namely online EM, Stochastic EM (SEM) and Data Augmentation (DA). To prevent the degeneracy inherent to all previous approaches (except [6]), the key point of our paper is to modify the contrast function to optimize. Instead of considering the likelihood function which leads to (5), we will consider here the so-called split-data likelihood (SDL) as originally proposed in [14], [15] for finite state-space HMM. In this approach, the data set is divided in blocks of say L data and one maximizes the average of the resulting log-SDL. This leads to an alternative Kullback-Leibler contrast function. It can be shown under regularity assumptions that the set of parameters optimizing this contrast function includes the true parameter. An approach consists of using the Fisher identity to obtain a gradient estimate. We will not discuss this approach here and we will maximize the average log-SDL using online EM type algorithms. As we work on data blocks of fixed dimension, the crucial point is that there is no more degeneracy problem. Moreover, contrary to [6], [12], [15], these algorithms are numerically well-behaved. An additional annealing schedule can also be added to the DA algorithm to make it more robust to initialization.

The rest of this paper is organized as follows. In Section 2, we introduce the average log-SPL. In Section 3, we present three recursive algorithms to optimize the resulting Kullback-Leibler contrast function and discuss the implementation issues. Finally in Section 4, we present an application to stochastic volatility.

2 Split-data likelihood

The standard likelihood function of $Y_{1:nL}$ is defined by

$$L_\theta(Y_{1:nL}) = \int \mu(x_0) \prod_{k=1}^{nL} f_\theta(x_{k-1}, x_k) \, g_\theta(x_k, Y_k) \, dx_{0:nL}.$$

The SDL consists of dividing the data $Y_{1:nL}$ in n blocks of L data. For each data block, the pseudo-likelihood $\widetilde{L}_\theta(Y_{(i-1)L+1:iL})$ is given by

$$\int \widetilde{p}_\theta\left(x_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right) dx_{(i-1)L+1:iL}, \quad (6)$$

with $\widetilde{p}_\theta\left(x_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right)$ equal to

$$\begin{aligned}
&\pi_\theta\left(x_{(i-1)L+1}\right) g_\theta\left(x_{(i-1)L+1}, Y_{(i-1)L+1}\right) \\
&\times \prod_{k=(i-1)L+2}^{iL} f_\theta(x_{k-1}, x_k) \, g_\theta(x_k, Y_k),
\end{aligned} \quad (7)$$

where π_θ correspond to the invariant density of the latent Markov process. The SDL corresponds to the product of the n pseudo-likelihoods (6). It is important to remark that our algorithms will require the knowledge of the analytical expression of this invariant density up to a normalizing constant. This is a restriction. However, this density is known in many important applications; it is satisfied for instance for all the examples addressed in [16].

We propose to maximize the average log-SDL

$$\frac{1}{n} \sum_{i=1}^{n} \widetilde{l}_\theta\left(Y_{(i-1)L+1:iL}\right) \rightarrow \widetilde{l}(\theta) \quad (8)$$

where $\widetilde{l}_\theta\left(Y_{(i-1)L+1:iL}\right) \triangleq \log \widetilde{L}_\theta\left(Y_{(i-1)L+1:iL}\right)$ and

$$\widetilde{l}(\theta) = \int \widetilde{l}_\theta(Y_{1:L}) \, p_{\theta^*}(Y_{1:L}) \, dY_{1:L},$$

$p_{\theta^*}(Y_{1:L})$ corresponds to the invariant distribution of the observations under the true parameter θ^*; i.e. this is the marginal of (7) for $i = 1$, $\theta = \theta^*$. Note the difference with the standard RML approach, see (4). Maximizing $\widetilde{l}(\theta)$ is equivalent to minimizing $\widetilde{K}(\theta, \theta^*) \triangleq \widetilde{l}(\theta^*) - \widetilde{l}(\theta)$ which satisfies under regularity assumptions $\widetilde{K}(\theta^*, \theta^*) = 0$ [14]. There is a trade-off associated to the choice of L. If L is small, the algorithm is typically easier to implement but the convergence of the algorithm might be slow. If L is large, the algorithm will converge faster as one mimics the convergence properties of the RML estimate but the algorithm is getting more complex.

3 Recursive Algorithms and Implementation

We describe briefly in this section the algorithms. A detailed example is given in the next section. In practice we consider only models for which the joint density $p_\theta(x_{0:n}, y_{1:n})$ is in the exponential family so that one only needs to propagate a set of sufficient statistics. We will denote by Φ the set of (typically multivariate) sufficient statistics. All algorithms rely on a non-increasing positive stepsize sequence $\{\gamma_i\}_{i\geq 0}$ satisfying $\sum \gamma_i = \infty$, $\sum \gamma_i^2 < \infty$; one usually selects $\gamma_i = i^{-\alpha}$ with $\alpha \in \left(\frac{1}{2}, 1\right]$.

3.1 Algorithms

The online EM algorithm proceeds as follows.

Online EM

- Initialization: $i = 0$, $\Phi^{(0)} = 0$ and $\theta^{(0)}$.
- Iteration i, $i \geq 1$
  - $\Phi^{(i)} = (1-\gamma_i)\,\Phi^{(i-1)}$ $+\gamma_i \widetilde{\mathbb{E}}_{\theta^{(i-1)}}\left(\varphi\left(X_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right)\middle| Y_{(i-1)L+1:iL}\right)$.
  - $\theta^{(i)} = \Psi\left(\Phi^{(i)}\right)$.

where $\widetilde{\mathbb{E}}_{\theta^{(i-1)}}\left(\varphi\left(X_{(i-1)L+1:iL}\right)\middle| Y_{(i-1)L+1:iL}\right)$ denotes the sufficient statistics associated to the data block $Y_{(i-1)L+1:iL}$ and Ψ is the mapping between the set of sufficient statistics and the parameter space. The symbol $\widetilde{\mathbb{E}}_{\theta^{(i-1)}}\left(\cdot\middle| Y_{(i-1)L+1:iL}\right)$ denotes the expectation with respect to $\widetilde{p}_\theta\left(x_{(i-1)L+1:iL}\middle| Y_{(i-1)L+1:iL}\right) \propto \widetilde{p}_\theta\left(x_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right)$.

The online SEM algorithm is a simple variation of the online EM which proceeds as follows.

Online SEM

- Initialization: $i = 0$, $\Phi^{(0)} = 0$ and $\theta^{(0)}$.
- Iteration i, $i \geq 1$
  - Sample $\widetilde{\widetilde{X}}_{(i-1)L+1:iL} \sim \widetilde{p}_{\theta^{(i-1)}}\left(x_{(i-1)L+1:iL}\middle| Y_{(i-1)L+1:iL}\right)$.
  - $\Phi^{(i)} = (1-\gamma_i)\,\Phi^{(i-1)} + \gamma_i \varphi\left(\widetilde{\widetilde{X}}_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right)$.
  - $\theta^{(i)} = \Psi\left(\Phi^{(i)}\right)$.

In the SEM algorithm, we replace the expectation term by an unbiased estimate. This has the effect of adding "noise" in the algorithm and can allow it to escape from a local maximum.

The online DA algorithm is a more recent variant introduced by the authors in [2]. In this case, one sets a prior density $p(\theta)$ on the unknown parameter θ and define the following artificial conditional density for the parameter θ

$$\widetilde{p}(\theta|\, x_{1:nL}, Y_{1:nL}) \propto \left[l(\theta; x_{1:nL}, Y_{1:nL})\right]^{\beta_n} p(\theta)$$

where $\{\beta_n\}_{n\geq 0}$ is the inverse of the "temperature" and $l(\theta; x_{1:nL}, Y_{1:nL})$ is defined recursively as follows

$$\begin{aligned} l(\theta; x_{1:L}, Y_{1:L}) &= \widetilde{p}_\theta(x_{1:L}, Y_{1:L}), \\ l(\theta; x_{1:iL}, Y_{1:iL}) &= \left[l\left(\theta; x_{1:(i-1)L}, Y_{1:(i-1)L}\right)\right]^{1-\gamma_i} \\ &\quad \times \left[\widetilde{p}_\theta\left(x_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right)\right]^{\gamma_i}. \end{aligned}$$

Though it is not entirely obvious, one can show that this algorithm corresponds to a "noisy" online EM algorithm in the case where $\beta_n = n$. The additional parameter β_n corresponds to an annealing schedule that can be used to slower the concentration of $\widetilde{p}(\theta|\, x_{1:nL}, Y_{1:nL})$; typically one will chose $\beta_n = An^\delta$ $(\delta > 0)$. For models of interest, $\widetilde{p}(\theta|\, x_{1:nL}, Y_{1:nL})$ only depends on a set of sufficient statistics (similar to those of the EM and SEM) and the online DA algorithm proceeds as follows.

Online DA

- Initialization: $i = 0$, $\Phi^{(0)} = 0$ and $\theta^{(0)}$.
- Iteration i, $i \geq 1$
  - Sample $\widetilde{\widetilde{X}}_{(i-1)L+1:iL} \sim \widetilde{p}_{\theta^{(i-1)}}\left(x_{(i-1)L+1:iL}\middle| Y_{(i-1)L+1:iL}\right)$.
  - $\Phi^{(i)} = (1-\gamma_i)\,\Phi^{(i-1)} + \gamma_i \varphi\left(\widetilde{\widetilde{X}}_{(i-1)L+1:iL}, Y_{(i-1)L+1:iL}\right)$.
  - Sample $\theta^{(i)} \sim \widetilde{p}\left(\theta\middle|\, \Phi^{(i)}\right)$.

Remark. The different components of the vector θ might be updated one-at-a-time using a Gibbs sampling strategy.

Remark. Note that instead of using non-overlapping data blocks $\{Y_{(i-1)L+1:iL}\}_{i\geq 1}$, it is possible to use a sliding window. This enables one to update the parameter estimate at the data rate for example [14].

3.2 Implementation Issues

The algorithms presented above assume that we know how to integrate with respect to $\widetilde{p}_\theta\left(x_{(i-1)L+1:iL}\middle| Y_{(i-1)L+1:iL}\right)$ or to sample from this density. It is typically impossible but one can perform these integration/sampling steps exactly or approximately using modern simulation techniques; i.e. Markov Chain Monte Carlo (MCMC) and SMC. When one uses SMC to estimate this density, the approximation one gets will be reasonable only if L is not too large. If not, one can use the forward filtering backward sampling algorithm proposed in [9] to sample from the joint density based on the particle approximations of the marginal filtering densities $\widetilde{p}_\theta\left(x_k\middle| Y_{(i-1)L+1:k}\right)$. One should keep in mind that even if one uses this method, the algorithm is still an online algorithm.

4 Application

Let us consider the following stochastic volatility model arising in finance

$$\begin{aligned} X_{n+1} &= \phi X_n + \sigma V_{n+1}, \\ Y_n &= \beta \exp(X_n/2)\, W_n, \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ are two mutually sequences, independent of the initial state X_0. We are interested in estimating the parameter $\theta \triangleq (\phi, \sigma, \beta)$. In this case, the stationary distribution of the hidden process is $\mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right)$. One has

$$\begin{aligned} & \log \widetilde{p}_\theta(x_{1:L}, Y_{1:L}) \\ = \; & cste - \frac{1}{2}\log\left(1-\phi^2\right) - L\log\sigma - L\log\beta \\ & - \frac{\left(1-\phi^2\right)}{2\sigma^2} x_1^2 - \frac{1}{2\beta^2}\sum_{k=1}^{L} Y_k^2 \exp(-x_k) \\ & - \frac{1}{2\sigma^2}\sum_{k=2}^{L}\left(x_k - \phi x_{k-1}\right)^2 \end{aligned}$$

so clearly $\varphi(x_{1:L}, Y_{1:L})$ is given by $\left(x_1^2, \sum_{k=1}^{L} Y_k^2 \exp(-x_k), \sum_{k=1}^{L-1} x_k^2, \sum_{k=2}^{L} x_k^2, \sum_{k=2}^{L} x_k x_{k-1}\right)$. If one writes $\Phi = (\Phi_1, \ldots, \Phi_5)$ then the mapping Ψ between the sufficient statistics and the parameter space necessary to the EM and SEM algorithms is defined as follows $\beta = \sqrt{\frac{1}{L}\Phi_2}$ and

$$\sigma^2 = \frac{1}{L}\left(\left(1-\phi^2\right)\Phi_1 + \phi^2\Phi_3 - 2\phi\Phi_5 + \Phi_4\right),$$
$$\left(1-\phi^2\right)^{-1}\phi + (\Phi_1 - \Phi_3)\,\sigma^{-2}\phi + \sigma^{-2}\Phi_5 = 0.$$

Plugging the expression of σ^2 in the last equation gives an equation in ϕ that can be solved by a simple dichotomic search. For the DA augmentation algorithm, we used a rejection technique to sample from $\widetilde{p}(\theta|\,\Phi)$. We present here a numerical experiment on synthetic data. The true parameter values are $\phi = 0.9, \beta = 1.0, \sigma^2 = 0.1$ and the RDA algorithm was used for $L = 2$ with a temperature $\beta_n = n^{0.5}$.

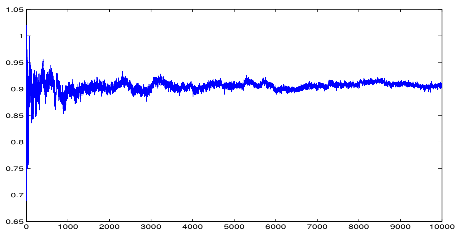

Figure 1: Convergence of the estimate of ϕ.

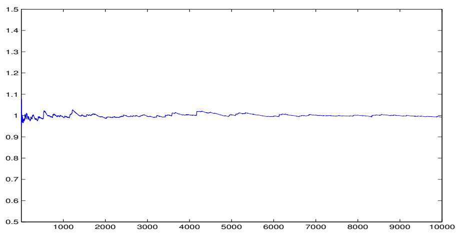

Figure 2: Convergence of the estimate of β.

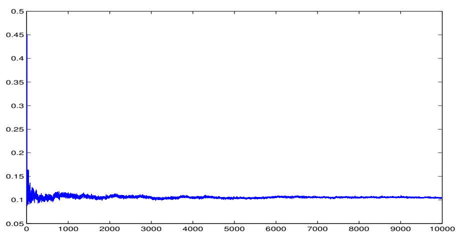

Figure 3: Convergence of the estimate of σ^2.

5 Discussion

In this paper, we have proposed three original algorithms to perform online parameter estimation in general state-space models. These algorithms are online EM type algorithms. Provided the invariant distribution of the hidden Markov process is known, these algorithms are simple and we have demonstrated their efficiency in practice. An alternative and computationally very efficient strategy which does not require state estimation has been recently proposed in [3].

6 Acknowledgments

This work was initiated while the second author was visiting the Department of Mathematics from Bristol University thanks to an EPSRC visiting fellowship.

7 REFERENCES

[1] Andrieu, C., De Freitas, J.F.G. and Doucet, A. (1999). Sequential MCMC for Bayesian model selection, *Proc. Work. IEEE HOS*.

[2] Andrieu, C. and Doucet A. (2000) Stochastic algorithms for marginal MAP retrieval of sinusoids in non-Gaussian noise, *Proc. Work. IEEE SSAP*.

[3] Andrieu, C., Doucet, A. and Tadić, V.B. (2003). Online sampling for parameter estimation in general state space models, *Proc. Worshop Sysid*, Rotterdam.

[4] Doucet, A., Godsill, S.J. and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statist. Comput.*, **10**, 197-208.

[5] Doucet, A., de Freitas, J.F.G. and Gordon N.J. (eds.) (2001). *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.

[6] Doucet A. & Tadić V.B. (2003) Parameter estimation in general state-space models using particle methods. *Ann. Inst. Stat. Math.*, to appear.

[7] Elliott R.J. & Krishnamurthy V. (1999) New finite dimensional filters for estimation of linear Gauss-Markov models, *IEEE Trans. Auto Control*, Vol.44, No.5, pp.938-951.

[8] Gilks, W.R. and Berzuini, C. (2001). Following a moving target - Monte Carlo inference for dynamic Bayesian models. *J. R. Statist. Soc.* B, **63**, pp. 127-146.

[9] Godsill, S.J., Doucet, A. & West M. (2000) Monte Carlo smoothing for nonlinear time series. Technical Report TR00-01, Institute Statistics and Decision Sciences, Duke University.

[10] Higuchi T. (1997) Monte Carlo filter using the genetic algorithm operators. *J. Statist. Comp. Simul.*, **59**, 1-23.

[11] Kitagawa, G. (1998). A self-organizing state-space model. *J. Am. Statist. Ass.*, **93**, 1203-1215.

[12] LeGland F. and Mevel L. (1997) Recursive identification in hidden Markov models, *Proc. 36th IEEE Conf. Decision and Control*, pp. 3468-3473.

[13] Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice* (Doucet, A., de Freitas, J.F.G. and Gordon N.J. (eds.)). New York: Springer-Verlag.

[14] Rydén T. (1994) Consistent and asymptotically normal parameter estimates for hidden Markov models. *Annals of Statistics* 22, 1884-1895.

[15] Rydén T. (1997) On recursive estimation for hidden Markov models. *Stochastic Processes and their Applications* 66, 79-96.

[16] Storvik G. (2002) Particle filters in state space models with the presence of unknown static parameters. *IEEE. Trans. Signal Processing*, 50, 281–289.