ONLINE SAMPLING FOR PARAMETER ESTIMATION IN GENERAL STATE SPACE MODELS

Christope Andrieu * **Arnaud Doucet** **
Vladislav B. Tadić ***

* *Department of Mathematics, Bristol University, BS8 1TW Bristol, UK*
** *Department of Engineering, Cambridge University, CB2 1PZ Cambridge, UK*
*** *Department of Electrical Engineering, Melbourne University, 3010 Melbourne, Australia*

Abstract: We consider the class of stationary nonlinear non Gaussian state space models with unknown static parameters. We propose original online stochastic gradient type algorithms to estimate these parameters. These algorithms rely on the simulation of artificial observations. Contrary to all the methods we are aware of in this framework, optimal state estimation is not required by our methods and the proposed algorithms are computationally efficient. Their efficiency is assessed through simulation.

Keywords: Parameter Estimation, Sequential Importance Sampling, State Space Models, Stochastic Approximation

1. INTRODUCTION

1.1 State Space Models and Problem Statement

Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 1}$ be R^p and R^q-valued stochastic processes defined on a measurable space $(\Omega, \mathcal{F})$ while $\theta \in \Theta$ where Θ is an open subset of R^d. 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 $f_\theta(\cdot,\cdot)$; i.e.

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

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

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

All densities are defined with respect to some appropriate dominating measures; e.g. Lebesgue. This class of models include 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 $\{V_n\}_{n\geq 0}$ and $\{W_n\}_{n\geq 1}$ are independent sequences and φ_θ, ψ_θ are deterministic functions.

We will restrict ourselves to stationarity state space models where, for any $\theta \in \Theta$, the Markov process $\{X_n\}_{n\geq 0}$ is ergodic and admits an invariant distribution π_θ.

Let us assume that the true value of the parameter θ is θ^* and that only the process $\{Y_n\}_{n\geq 1}$ is observed. We are interested in deriving recursive algorithms to estimate θ^*. This complex problem has numerous applications in electrical engineering, econometrics, statistics, etc.

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

Following the introduction of Sequential Monte Carlo (SMC) methods (Doucet, *et al.*, 2001), i.e. particle filters, many methods have been recently proposed to address this problem. There are roughly three categories of methods.

• *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 $S_n = (X_n, \theta)$. This converts the parameter estimation into an optimal filtering problem. One can then apply, at least theoretically, standard particle filtering techniques (Doucet, *et al.*, 2001) to estimate the joint posterior density $p(x_n, \theta | Y_{1:n})$ 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. 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 model on the parameter of interest; see (Higuchi, 1997), (Kitagawa, 1998). To avoid the introduction of such a model, an approach proposed in (Gilks and Berzuini, 2001) consists of adding Markov chain Monte Carlo (MCMC) steps so as to add "diversity" among the particles. However, this approach does not really 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.

• *Recursive Maximum Likelihood*, RML is a stochastic gradient type algorithm to maximize the average log-likelihood. This approach requires the computation of the optimal filter $p_\theta(x_n | Y_{1:n})$ and its derivative with respect to θ. This is the approach followed in (LeGland and Mevel, 1997) for finite state space HMM. Algorithms to compute numerically the derivative for general state space models have been proposed in (Cérou, et al., 2001) and (Doucet and Tadić, 2003).

• *Online Expectation-Maximization (EM).* Online EM is an alternative stochastic gradient type algorithm. Such algorithms have been proposed for finite state space HMM and linear Gaussian state space models. It is formally possible to come up with a similar algorithm for general state space models. However it requires the computation of an (online) approximation of the joint density $p_\theta(x_{1:n} | Y_{1:n})$ whose dimension increases with time. To avoid this problem, one can use the split data likelihood (Rydén, 1994); see (Andrieu and Doucet, 2003).

All these methods rely on computing non standard posterior distributions and thus require the use of numerical techniques such as SMC methods. For some applications, SMC methods are still too computationally intensive. This has motivated the development of computationally cheaper methods.

1.3 Contributions

We propose here alternative algorithms to address the problem of recursive parameter estimation in general state space models. These algorithms rely on the introduction of a non standard cost function (i.e. different from the classical Kullback-Leibler cost function) which can be shown to be minimized for θ^*. The main difference between our algorithms and all other algorithms we are aware of is that state estimation is not required in our framework. The algorithms are computationally several orders of magnitude cheaper than algorithms based on SMC methods.

The rest of this paper is organized as follows. In Section 2, we introduce the cost function to maximize. In Section 3, we present recursive algorithms to optimize this cost function and discuss the implementation issues. Finally in Section 4, we present an application to a stochastic volatility model arising in econometrics.

2. STATIONARY DISTRIBUTION AND COST FUNCTION

In the stationary regime, the distribution of the random variables $\left(X_{(k-1)L+1:kL}, Y_{(k-1)L+1:kL}\right)$ satisfies

$$P_\theta\left(\left(X_{(k-1)L+1:kL}, Y_{(k-1)L+1:kL}\right) \in d(x_{1:L}, y_{1:L})\right) = \pi_\theta(x_{1:L}, y_{1:L})\, dx_{1:L} dy_{1:L}$$

where $d(\cdot)$ is the dominating measure and

$$\pi_\theta(x_{1:L}, y_{1:L}) = \pi_\theta(x_1)\, g_\theta(y_1 | x_1) \times \prod_{i=2}^{L} f_\theta(x_i | x_{i-1})\, g_\theta(y_i | x_i)$$

where we recall that π_θ corresponds to the invariant density of the latent Markov process.

We will consider the following cost function

$$J(\theta) = \int \left(\pi_\theta(y_{1:L}) - \pi_{\theta^*}(y_{1:L})\right)^2 dy_{1:L} \qquad (3)$$

where

$$\pi_\theta(y_{1:L}) = \int \pi_\theta(x_{1:L}, y_{1:L})\, dx_{1:L}.$$

The cost function $J(\theta)$ is defined under weak assumptions and it is clear that θ^* minimizes it. The parameter L is chosen large enough to ensure identifiability of θ using $\pi_\theta(y_{1:L})$.

3. ONLINE SAMPLING ALGORITHMS

We will distinguish here two cases. If the invariant distribution π_θ is known, then one can devise a standard stochastic gradient type algorithm to minimize $J(\theta)$. If this distribution is unknown, then it is necessary to use a (randomized) finite difference scheme.

3.1 Invariant distribution known

We describe briefly in this section a stochastic gradient algorithm to minimize $J(\theta)$. One has

$$J(\theta) = \int \pi_\theta(y_{1:L})\left(\pi_\theta(y_{1:L}) - 2\pi_{\theta^*}(y_{1:L})\right) dy_{1:L} + \text{terms independent of } \theta.$$

The derivative $\frac{1}{2}\nabla J(\theta)$ of the cost function is given by

$$\int \nabla\pi_\theta(y_{1:L})\left(\pi_\theta(y_{1:L}) - \pi_{\theta^*}(y_{1:L})\right) dy_{1:L}.$$

Under regularity conditions, one has

$$\begin{aligned}\nabla\pi_\theta(y_{1:L}) &= \nabla\textstyle\int \pi_\theta(x_{1:L}, y_{1:L})\, dx_{1:L}\\ &= \textstyle\int \left[\nabla \log \pi_\theta(x_{1:L}, y_{1:L})\right] \pi_\theta(x_{1:L}, y_{1:L})\, dx_{1:L}\\ &= \textstyle\int \left[\nabla \log \pi_\theta(x_{1:L}, y_{1:L})\right] g_\theta(y_{1:L}|\, x_{1:L})\, \pi_\theta(x_{1:L})\, dx_{1:L}\end{aligned}$$

where one uses the following notation $\pi_\theta(x_{1:L}, y_{1:L}) = g_\theta(y_{1:L}|\, x_{1:L})\, \pi_\theta(x_{1:L})$ with

$$g_\theta(y_{1:L}|\, x_{1:L}) = \prod_{i=1}^{L} g_\theta(y_i|\, x_i)$$

to emphasize that this density does not depend on on the invariant density of $\{X_n\}_{n\geq 0}$.

To sum up, one obtains for $\frac{1}{2}\nabla J(\theta)$

$$\int \nabla \log \pi_\theta(x_{1:L}, y_{1:L}) \times g_\theta(y_{1:L}|\, x_{1:L})\, \pi_\theta(x_{1:L}) \times \left(\pi_\theta(y_{1:L}) - \pi_{\theta^*}(y_{1:L})\right) dx_{1:L} dy_{1:L}. \quad (4)$$

Let $X^*_{(k-1)L+1:kL}$ denote a realization of $\pi_{\theta_k}(x_{1:L})$, $Y^*_{(k-1)L+1:kL}$ a realization from $\pi_{\theta_k}(y_{1:L})$ (the variables $X^*_{(k-1)L+1:kL}$ and $Y^*_{(k-1)L+1:kL}$ being statistically independent) and $Y_{(k-1)L+1:kL}$ a realization from $\pi_{\theta^*}(y_{1:L})$ (i.e. the available data), an asymptotically [1] unbiased gradient estimate of (4) for $\theta = \theta_k$ is given by

$$\begin{aligned}&\tfrac{1}{2}\widehat{\nabla} J(\theta_k) =\\ &\nabla \log \pi_{\theta_k}\left(X^*_{(k-1)L+1:kL}, Y^*_{(k-1)L+1:kL}\right)\\ &\times g_{\theta_k}\left(Y^*_{(k-1)L+1:kL}\,\middle|\, X^*_{(k-1)L+1:kL}\right)\\ &-\nabla \log \pi_{\theta_k}\left(X^*_{(k-1)L+1:kL}, Y_{(k-1)L+1:kL}\right)\\ &\times g_{\theta_k}\left(Y_{(k-1)L+1:kL}\,\middle|\, X^*_{(k-1)L+1:kL}\right)\end{aligned} \quad (5)$$

The stochastic gradient algorithm to minimize $J(\theta)$ follows. It relies on a non-increasing positive stepsize sequence $\{\gamma_k\}_{k\geq 0}$ satisfying $\sum \gamma_k = \infty$, $\sum \gamma_k^2 < \infty$; one usually selects $\gamma_k \propto k^{-\alpha}$ with $\alpha \in \left(\frac{1}{2}, 1\right]$.

Sampling step

- Sample $\widetilde{X}_{(k-1)L+1:kL} \sim \pi_{\theta_k}(\cdot)$,

$$Y^*_{(k-1)L+1:kL} \sim g_{\theta_k}\left(\cdot|\, \widetilde{X}_{(k-1)L+1:kL}\right).$$

- Sample $X^*_{(k-1)L+1:kL} \sim \pi_{\theta_k}(\cdot)$.

Gradient estimation step

- Compute $\frac{1}{2}\widehat{\nabla} J(\theta_k)$ using (5).

Parameter updating step

$$\theta_{k+1} = \theta_k - \gamma_{k+1}\widehat{\nabla} J(\theta_k).$$

Remark. A truly recursive algorithm, i.e. updating the parameter estimate at each time step, can be derived easily by using a "sliding" window instead of partitioning the data in separated blocks.

Note that $\widetilde{X}_{(k-1)L+1:kL}$ and $X^*_{(k-1)L+1:kL}$ need to be statistically independent to ensure that $X^*_{(k-1)L+1:kL}$ and $Y^*_{(k-1)L+1:kL}$ are independent. Contrary to all algorithms we are aware of, this algorithm only requires the simulation of variables according to the prior distribution.

It is possible to reduce the variance of the gradient estimate by using Sequential Importance Sampling (SIS) techniques (Doucet, *et al.*, 2001); i.e. instead of imputing the latent process according to its (stationary) prior distribution one can come up with "clever" importance distributions to reduce the variance of the gradient. One can also avoid sampling from the invariant distribution and use instead the following algorithm.

Sampling step

- Sample $\widetilde{X}_{(k-1)L+1:kL} \sim f_{\theta_k}\left(\cdot|\, \widetilde{X}_{(k-1)L}\right)$,

$$Y^*_{(k-1)L+1:kL} \sim g_{\theta_k}\left(\cdot|\, \widetilde{X}_{(k-1)L+1:kL}\right).$$

---

[1] The true system needs to reach its stationary regime.

• Sample $X^*_{(k-1)L+1:kL} \sim f_{\theta_k}\left(\cdot \middle| X^*_{(k-1)L}\right)$.

Gradient estimation step

• Compute $\frac{1}{2}\widehat{\nabla} J(\theta_k)$ using (5).

Parameter updating step

$$\theta_{k+1} = \theta_k - \gamma_{k+1}\widehat{\nabla} J(\theta_k).$$

In the above algorithm, one uses the following notation

$$f_\theta\left(x_{(k-1)L+1:kL} \middle| x_{(k-1)L}\right) = \prod_{i=(k-1)L+1}^{kL} f_\theta\left(x_i \middle| x_{i-1}\right).$$

It is important to remark that even if one can avoid sampling directly from the invariant distribution, one needs to know it analytically to compute the gradient estimate. This is a restriction even if this density is known in many important applications; e.g. state space models with a linear Gaussian evolution equation. In cases where this density is not known, one needs to use an alternative algorithm developed in the following subsection.

3.2 Invariant distribution unknown

We propose here another stochastic approximation method namely SPSA (Simultaneous Perturbation Stochastic Approximation) due to Spall (Spall, 1998) as an alternative way to optimize the cost function. With SPSA, the gradient is approximated via a finite difference method using only the estimates of the cost function of interest. This technique has proved successful among other finite difference methods as it only requires two estimates of the cost function regardless of the dimension d of the parameter to obtain a gradient estimate. The SPSA technique requires all elements of θ to be varied randomly simultaneously to obtain two estimates of the cost function. The two estimates required are of the form $J(\theta \pm \text{perturbation})$ for a two-sided gradient approximation. In this case, the gradient estimate $\widehat{\nabla} J(\theta_k) = \left(\widehat{\nabla} J_1(\theta_k), \ldots, \widehat{\nabla} J_d(\theta_k)\right)^{\mathrm{T}}$ is given by

$$\widehat{\nabla} J_i(\theta_k) = \frac{\widehat{J}(\theta_k + c_k\Delta_k) - \widehat{J}(\theta_k - c_k\Delta_k)}{2c_k\Delta_{k,i}}$$

where $\{c_k\}_{k\geq 1}$ denotes a sequence of positive scalars such that $c_k \to 0$ and

$$\Delta_k = (\Delta_{k,1}, \Delta_{k,2}, \cdots, \Delta_{k,d})$$

is a d-dimensional random perturbation vector. Careful selection of algorithm parameters γ_n, c_n and Δ_n is required to ensure convergence. The $\{\gamma_k\}_{k\geq 1}$ and $\{c_k\}_{k\geq 1}$ sequences generally take the form of $\gamma_k \propto k^{-\alpha}$ and $c_k \propto k^{-\beta}$ respectively with non-negative coefficients α and $\beta \in \left(\frac{1}{2}, 1\right]$. Each component of Δ_k is usually generated from a symmetric Bernoulli ± 1 distribution. See (Spall, 1998) for guidelines on coefficient selection.

To obtain an estimate of $J(\theta)$ for a given value, one recalls that

$$J(\theta) = \int \pi_\theta(y_{1:L})\left(\pi_\theta(y_{1:L}) - \pi_{\theta^*}(y_{1:L})\right) dy_{1:L} + \text{terms independent of } \theta.$$

and for any θ', θ''

$$\int \pi_{\theta'}(y_{1:L})\,\pi_{\theta''}(y_{1:L})\,dy_{1:L} = \int\int g_{\theta'}(y_{1:L}|x_{1:L})\,\pi_{\theta'}(x_{1:L})\,\pi_{\theta''}(y_{1:L})\,dx_{1:L}dy_{1:L}.$$

It follows that by sampling a realization $X^+_{(k-1)L+1:kL}$ (resp. $X^-_{(k-1)L+1:kL}$) from $\pi_{\theta_k+c_k\Delta_k}(x_{1:L})$ (resp. from $\pi_{\theta_k-c_k\Delta_k}(x_{1:L})$), a realization $Y^+_{(k-1)L+1:kL}$ (resp. $Y^-_{(k-1)L+1:kL}$) from $\pi_{\theta_k+c_k\Delta_k}(y_{1:L})$ (resp. from $\pi_{\theta_k+c_k\Delta_k}(y_{1:L})$) and $Y_{(k-1)L+1:kL}$ a realization from $\pi_{\theta^*}(y_{1:L})$ (i.e. the data available), one obtains an (asymptotically) unbiased estimate of

$$\frac{1}{2}\left(J(\theta_k + c_k\Delta_k) - J(\theta_k - c_k\Delta_k)\right)$$

using

$$\begin{aligned}&\tfrac{1}{2}\left(\widehat{J}(\theta_k + c_k\Delta_k) - \widehat{J}(\theta_k - c_k\Delta_k)\right) = \\ &g_{\theta_k+c_k\Delta_k}\left(Y^+_{(k-1)L+1:kL} \middle| X^+_{(k-1)L+1:kL}\right) \\ &-g_{\theta_k+c_k\Delta_k}\left(Y_{(k-1)L+1:kL} \middle| X^+_{(k-1)L+1:kL}\right) \\ &-g_{\theta_k-c_k\Delta_k}\left(Y^-_{(k-1)L+1:kL} \middle| X^-_{(k-1)L+1:kL}\right) \\ &+g_{\theta_k-c_k\Delta_k}\left(Y_{(k-1)L+1:kL} \middle| X^-_{(k-1)L+1:kL}\right).\end{aligned} \tag{6}$$

Note that one needs $X^+_{(k-1)L+1:kL}$ (resp. $X^-_{(k-1)L+1:kL}$) to be statistically independent from $Y^+_{(k-1)L+1:kL}$ (resp. $Y^-_{(k-1)L+1:kL}$). However $X^+_{(k-1)L+1:kL}$ (resp. $X^-_{(k-1)L+1:kL}$) can be statistically dependent from $X^-_{(k-1)L+1:kL}$ (resp. $X^+_{(k-1)L+1:kL}$) and $Y^-_{(k-1)L+1:kL}$ (resp. $Y^+_{(k-1)L+1:kL}$).

The algorithm to minimize $J(\theta)$ follows.

Sampling step

• Sample $\Delta_k = (\Delta_{k,1}, \ldots, \Delta_{k,d})$.

• Sample $X^+_{(k-1)L+1:kL} \sim \pi_{\theta_k+c_k\Delta_k}(\cdot)$

• Sample $X^-_{(k-1)L+1:kL} \sim \pi_{\theta_k-c_k\Delta_k}(\cdot)$.

• Sample $\widetilde{X}^+_{(k-1)L+1:kL} \sim \pi_{\theta_k+c_k\Delta_k}(\cdot)$,

$Y^+_{(k-1)L+1:kL} \sim g_{\theta_k+c_k\Delta_k}\left(\cdot \middle| \widetilde{X}^+_{(k-1)L+1:kL}\right)$.

• Sample $\widetilde{X}^-_{(k-1)L+1:kL} \sim \pi_{\theta_k-c_k\Delta_k}(\cdot)$,

$Y^-_{(k-1)L+1:kL} \sim g_{\theta_k-c_k\Delta_k}\left(\cdot \middle| \widetilde{X}^-_{(k-1)L+1:kL}\right)$.

Gradient estimation step

- Compute $\frac{1}{2}\left(\widehat{J}(\theta_k + c_k\Delta_k) - \widehat{J}(\theta_k - c_k\Delta_k)\right)$ using (6).
- For $i = 1$ to d, evaluate gradient components as

$$\widehat{\nabla} J_i(\theta_k) = \frac{\widehat{J}(\theta_k + c_k\Delta_k) - \widehat{J}(\theta_k - c_k\Delta_k)}{2c_k\Delta_{k,i}}$$

Parameter updating step

$$\theta_{k+1} = \theta_k - \gamma_{k+1}\widehat{\nabla} J(\theta_k).$$

The problem is that it is typically impossible to sample exactly from the distributions $\pi_{\theta_k + c_k\Delta_k}$ and $\pi_{\theta_k - c_k\Delta_k}$. However one can use the following algorithm instead.

Sampling step

- Sample $\Delta_k = (\Delta_{k,1}, \ldots, \Delta_{k,d})$.
- Sample $X^+_{(k-1)L+1:kL} \sim f_{\theta_k + c_k\Delta_k}\left(\cdot \mid X^+_{(k-1)L}\right)$
- Sample $X^-_{(k-1)L+1:kL} \sim f_{\theta_k - c_k\Delta_k}\left(\cdot \mid X^-_{(k-1)L}\right)$.
- Sample $\widetilde{X}^+_{(k-1)L+1:kL} \sim f_{\theta_k + c_k\Delta_k}\left(\cdot \mid \widetilde{X}^+_{(k-1)L}\right)$, $Y^+_{(k-1)L+1:kL} \sim g_{\theta_k + c_k\Delta_k}\left(\cdot \mid \widetilde{X}^+_{(k-1)L+1:kL}\right)$.
- Sample $\widetilde{X}^-_{(k-1)L+1:kL} \sim f_{\theta_k - c_k\Delta_k}\left(\cdot \mid \widetilde{X}^-_{(k-1)L}\right)$, $Y^-_{(k-1)L+1:kL} \sim g_{\theta_k - c_k\Delta_k}\left(\cdot \mid \widetilde{X}^-_{(k-1)L+1:kL}\right)$.

Gradient estimation step

- Compute $\frac{1}{2}\left(\widehat{J}(\theta_k + c_k\Delta_k) - \widehat{J}(\theta_k - c_k\Delta_k)\right)$ using (6).
- For $i = 1$ to d, evaluate gradient components as

$$\widehat{\nabla} J_i(\theta_k) = \frac{\widehat{J}(\theta_k + c_k\Delta_k) - \widehat{J}(\theta_k - c_k\Delta_k)}{2c_k\Delta_{k,i}}$$

Parameter updating step

$$\theta_{k+1} = \theta_k - \gamma_{k+1}\widehat{\nabla} J(\theta_k).$$

To reduce the variance of the gradient, SIS techniques can also be used in this case. Further variance reduction can be achieved by using common random numbers so as to introduce correlations between $X^+_{(k-1)L+1:kL}$ and $X^-_{(k-1)L+1:kL}$ or/and $Y^+_{(k-1)L+1:kL}$ and $Y^-_{(k-1)L+1:kL}$ (Kleinman, *et al.*, 1999).

4. APPLICATION

We demonstrate our methodology on a non linear state space model. Let us consider the following stochastic volatility model arising in econometrics

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

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 independent sequences of independent identically distributed (i.i.d.) Gaussian random variables, independent of the initial state X_0. We are interested in estimating the parameter $\theta = (\beta, \phi, \sigma)$ where $\Theta = (0,1) \times (0,M) \times (0,M)$ with $M = 100$. In this case, the stationary distribution of the hidden process is $\mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right)$. We simulate 10000 observations with $\theta^* = (1, 0.8, 1)$. The algorithm is using $L = 2$.

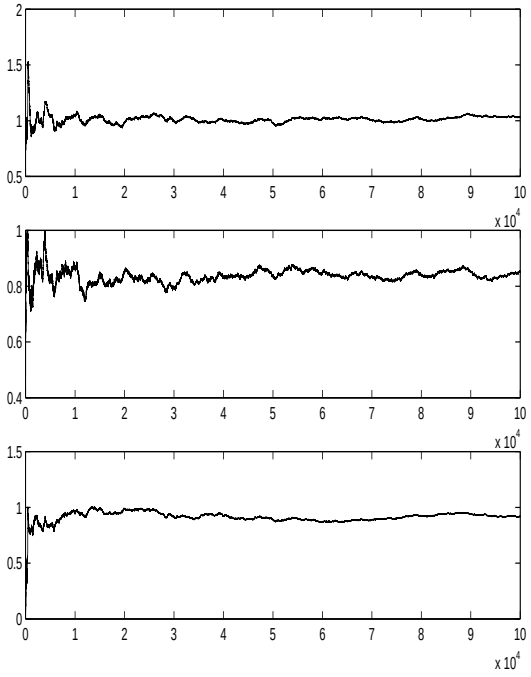

Fig. 1. Sequence of parameter estimates $\theta_n = (\beta_n, \phi_n, \sigma_n)$ for $N = 10000$. From top to bottom: β_n, ϕ_n and σ_n.

5. DISCUSSION

We have proposed new algorithms to perform parameter estimation in general state space models. These algorithms do not perform state estimation and are computationally efficient. Algorithmically, variance reduction techniques can be developed to improve their performance. Theoretically, our algorithms are "standard" stochastic gradient algorithms for which convergence results will be reported elsewhere.

6. ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council, UK.

7. REFERENCES

Andrieu C., and Doucet A. (2003). On-line Expectation-Maximization type algorithms for parameter estimation in general state space models. *Proc. IEEE Conf. ICASSP.*

Cérou F., LeGland F. and Newton N.J. (2001) Stochastic particle methods for linear tangent equations. in *Optimal Control and PDE's - Innovations and Applications* (eds. J. Menaldi, E. Rofman and A. Sulem), pp. 231-240, IOS Press, Amsterdam.

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

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

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.

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

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

Kleinman N.L., Spall J.C. and Naiman D.Q. (1999) Simulation-based optimisation with stochastic approximation using common random numbers. *Management Science*, **45**, pp. 1570-1578.

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

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

Spall J.C. (1998) Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimisation. *IEEE Trans. Aerospace Elect. Sys.*, **34**, 817-823.