Parameter Estimation in General State-Space Models using Particle Methods

Arnaud Doucet *

Signal Processing Group, Department of Engineering,

Cambridge University,

Trumpington Street, CB2 1PZ Cambridge, UK.

Email: ad2@eng.cam.ac.uk

Vladislav B. Tadić

Department of Electrical and Electronic Engineering,

University of Melbourne,

Parkville, Victoria 3052, Australia.

Email: v.tadic@ee.mu.oz.au

ABSTRACT

Particle filtering techniques are a set of powerful and versatile simulation-based methods to perform optimal state estimation in nonlinear non-Gaussian state-space models. If the model includes fixed parameters, a standard technique to perform parameter estimation consists of extending the state with the parameter to transform the problem into an optimal filtering problem. However, this approach requires the use of special particle filtering techniques which suffer from several drawbacks. We consider here an alternative approach combining particle filtering and gradient algorithms to perform batch and recursive maximum likelihood parameter estimation. An original particle method is presented to implement these approaches and their

*corresponding author.

efficiency is assessed through simulation.

***Keywords*: Optimal Filtering, Parameter Estimation, Sequential Monte Carlo, State-Space Models, Stochastic Approximation.**

I. Introduction

A. State-Space Models and Problem Statement

Let $(\Omega, \mathcal{F})$ be a measurable space. Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 0}$ be $\mathbb{R}^p$ and $\mathbb{R}^q$-valued stochastic processes defined on $(\Omega, \mathcal{F})$ and $\theta \in \Theta$ where Θ is an open subset of $\mathbb{R}^k$. There exist probability measures $\mathcal{P}_\theta : \mathcal{F} \to [0,1]$, $\mu : \mathcal{B}^p \to [0,1]$ and Borel-measurable functions $p_\theta : \mathbb{R}^p \times \mathbb{R}^p \to [0,\infty)$, $\varepsilon_\theta : \mathbb{R}^p \times \mathbb{R}^q \to [0,\infty)$ such that $\int p_\theta(x,x')\,dx' = \int \varepsilon_\theta(x,y)\,dy = 1$, $\mathcal{P}_\theta(X_0 \in B) = \int_B \mu(dx_0)$ and

$$\mathcal{P}_\theta\left(X_{n+1} \in B \middle| X^n, Y^n\right) = \int_B p_\theta(X_n, x)\,dx,\ \forall B \in \mathcal{B}^p,\ n \geq 0, \tag{1}$$

$$\mathcal{P}_\theta\left(Y_{n+1} \in B \middle| X^{n+1}, Y^n\right) = \int_B \varepsilon_\theta(X_{n+1}, y)\,dy,\ \forall B \in \mathcal{B}^q,\ n \geq 0, \tag{2}$$

where we denote by $Z^n \triangleq (Z_0, Z_2, ..., Z_n)$ the path of a process $\{Z_n\}_{n\geq 0}$ from time 0 to time n. This class of models include many nonlinear and non-Gaussian time series models [15].

Let us assume that the true value of the parameter θ is θ^* and that only the process $\{Y_n\}_{n\geq 0}$ is observed. We are interested in deriving a batch and a recursive algorithm to estimate θ^*. These problem are very complex. Indeed, even if θ^* were known, the optimal filtering problem, i.e. estimating the conditional distribution of X_n given Y^n, does not admit a closed-form solution and one needs to perform numerical approximations.

B. A Brief Literature Review

Recently, there has been a surge of interest in particle methods to perform optimal filtering in general state-space models; see [5] for a booklength survey and [12] for alternative applications of particle methods. Although particle filters have proven to be successful in many applications [5], parameter estimation using particle methods still remains a major problem. A standard approach followed in the literature consists of setting a prior distribution on the unknown parameter θ and then considering the extended state $Z_n \triangleq (X_n, \theta)$. This converts the parameter estimation into an optimal filtering problem. One can then apply, at least theoretically, standard particle filtering techniques. In this approach, if one were to use say the bootstrap filter [8], [13], then 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 [8], [18]. 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], [14]. In this case, we obtain a so-called self-organizing or self-tuning state-space model [14]. To avoid the introduction of an artificial dynamic, a clever approach proposed in [7] 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 as observed in [1]. Methods based on MCMC steps also require to be able to express the conditional distribution of the parameter given the data $\{Y_n\}_{n\geq 0}$ and the hidden process $\{X_n\}_{n\geq 0}$ through a set of fixed dimensional sufficient statistics. If this assumption is not satisfied, the computational complexity and memory requirements of these algorithms at time n are of order $\mathcal{O}(nN)$ for $N \gg 1$ particles; this makes these methods useless from a practical point of view.

We propose here two approaches to perform parameter estimation in general state-space models. The first method is a batch gradient algorithm to perform Maximum Likelihood (ML) parameter estimation where one computes at each iteration the gradient of the log-likelihood with respect to the parameters of interest. The second algorithm is a natural recursive version of the first one usually referred to as Recursive ML (RML). Loosely speaking, RML is a stochastic gradient method for maximizing the average log-likelihood. It has been developed originally in automatic control and signal processing where it is widely used; see for example [19]. In these approaches, it is not necessary to set a prior on the unknown static parameter and the conditional distribution of the parameter given the data and the hidden process can be arbitrary. In the context of general state-space models, these methods require essentially the ability to compute the optimal filter and the derivative of this filter with respect to the parameter of interest. We present here an original particle method to approximate the derivative of the filter and we apply it to perform batch and recursive ML.

C. Organization of the paper

The rest of the paper is organized as follows: In Section II, we detail the batch and recursive maximum likelihood approaches. Section III describes a particle method to implement the algorithms given in the previous section. In Section IV, we apply these algorithms to several parameter estimation problems.

II. Batch and Recursive Maximum Likelihood Estimation

For ease of presentation, we will consider the case where $\theta \in \mathbb{R}$ in the two following sections; the extension to a multidimensional parameter is straightforward and is described briefly in Subsection III-B.

The likelihood of the observations Y^n satisfies

$$L_\theta\left(Y^n\right) = \int \cdots \int \mu\left(dx_0\right) \prod_{k=1}^{n} \varepsilon_\theta\left(x_k, Y_k\right) \mathcal{P}_\theta\left(X_k \in dx_k | Y^{k-1}\right),$$

where $\mathcal{P}_\theta\left(X_k \in dx_k | Y^{k-1}\right)$ is the posterior distribution of the state X_k given the observations Y^{k-1} and θ; the true parameter value being θ^*. The log-likelihood function satisfies

$$l_\theta\left(Y^n\right) = \log\left(L_\theta\left(Y^n\right)\right) = \sum_{k=1}^{n} \log\left(\int \varepsilon_\theta\left(x_k, Y_k\right) \mathcal{P}_\theta\left(X_k \in dx_k | Y^{k-1}\right)\right).$$

We propose here two gradient type algorithms to perform ML estimation. The first method is a batch algorithm designed to maximize $l_\theta\left(Y^n\right)$. The second method is a recursive version of the first method maximizing $\lim_{n\to\infty} \frac{1}{n} l_\theta\left(Y^n\right)$.

A. Batch Maximum Likelihood

Let us assume the data Y^n are available. We want to maximize the log-likelihood $l_\theta\left(Y^n\right)$ using a gradient algorithm. At iteration $m+1$, the parameter estimate is updated through

$$\theta_{m+1} = \theta_m + \gamma_m \left(\partial l\right)_{\theta_m}\left(Y^n\right). \tag{3}$$

where ∂ denotes the derivative evaluated at the point θ_m. The stepsize sequence $\left\{\gamma_m\right\}_{m\geq 0}$ is a non-increasing sequence typically chosen as $\gamma_m = \gamma_0 . m^{-\alpha}$ with $\gamma_0 > 0$ and $0.5 < \alpha \leq 1$.

The derivative of the log-likelihood satisfies

$$\begin{aligned}
(\partial l)_\theta (Y^n) &= \sum_{k=1}^{n} \partial \log \left(\int \varepsilon_\theta (x_k, Y_k) \mathcal{P}_\theta \left(X_k \in dx_k | Y^{k-1} \right) \right) \\
&= \sum_{k=1}^{n} \frac{\int (\partial \varepsilon)_\theta (x_k, Y_k) \mathcal{P}_\theta \left(X_k \in dx_k | Y^{k-1} \right) + \int \varepsilon_\theta (x_k, Y_k) (\partial \mathcal{P})_\theta \left(X_k \in dx_k | Y^{k-1} \right)}{\int \varepsilon_\theta (x_k, Y_k) \mathcal{P}_\theta \left(X_k \in dx_k | Y^{k-1} \right)}.
\end{aligned} \tag{4}$$

The expression (4) involves both the one-step ahead prediction distribution $\mathcal{P}_\theta \left(X_k \in dx_k | Y^{k-1} \right)$ and its derivative $(\partial \mathcal{P})_\theta \left(X_k \in dx_k | Y^{k-1} \right)$. We are interested in computing recursively these quantities. Further on, we use the notation: $\pi^\theta_{k|l} (dx_k) \triangleq \mathcal{P}_\theta \left(X_k \in dx_k | Y^l \right)$ and $w^\theta_{k|l} (dx_k) = (\partial \mathcal{P})_\theta \left(X_k \in dx_k | Y^l \right)$. It follows that the derivative of the log-likelihood (4) can be rewritten as

$$(\partial l)_\theta (Y^n) = \sum_{k=1}^{n} \frac{\int (\partial \varepsilon)_\theta (x_k, Y_k) \pi^\theta_{k|k-1} (dx_k) + \int \varepsilon_\theta (x_k, Y_k) w^\theta_{k|k-1} (dx_k)}{\int \varepsilon_\theta (x_k, Y_k) \pi^\theta_{k|k-1} (dx_k)}. \tag{5}$$

For any measure $\mu(dx)$, transition kernel $g(x, dx')$ (not necessarily a probability measure or a transition probability kernel) and function $f(x)$, we also use the following standard notation

$$\mu g (dx') = \int \mu (dx) g (x, dx'), \ \langle \mu, f \rangle = \int \mu (dx) f (x), \ \mu.f (dx) = \mu (dx) f (x).$$

The probability distribution $\pi^\theta_{k|k-1}$ and the signed measure $w^\theta_{k|k-1}$ satisfy

$$\begin{aligned}
\pi^\theta_{k|k-1} &= \pi^\theta_{k-1|k-1} p_\theta, \ \pi^\theta_{k|k-1} (dx_k) = \int \pi^\theta_{k-1|k-1} (dx_{k-1}) p_\theta (x_{k-1}, x_k) dx_k, \\
\pi^\theta_{k|k} &= \frac{\pi^\theta_{k|k-1}.\varepsilon_\theta}{\left\langle \pi^\theta_{k|k-1}, \varepsilon_\theta \right\rangle}, \ \pi^\theta_{k|k} (dx_k) = \frac{\pi^\theta_{k|k-1} (dx_k) \varepsilon_\theta (x_k, Y_k)}{\int \pi^\theta_{k|k-1} (dx_k) \varepsilon_\theta (x_k, Y_k)},
\end{aligned}$$

and, using a similar notation,

$$\begin{aligned}
w^\theta_{k|k-1} &= w^\theta_{k-1|k-1} p_\theta + \pi^\theta_{k-1|k-1} (\partial p)_\theta, \\
w^\theta_{k|k} &= \frac{w^\theta_{k|k-1}.\varepsilon_\theta + \pi^\theta_{k|k-1}.(\partial \varepsilon)_\theta}{\left\langle \pi^\theta_{k|k-1}, \varepsilon_\theta \right\rangle} - \pi^\theta_{k|k} \frac{\left\langle w^\theta_{k|k-1}, \varepsilon_\theta \right\rangle + \left\langle \pi^\theta_{k|k-1}, (\partial \varepsilon)_\theta \right\rangle}{\left\langle \pi^\theta_{k|k-1}, \varepsilon_\theta \right\rangle}.
\end{aligned}$$

As $w^\theta_{k|l} (dx_k)$ is the derivative of a probability measure, one can check that under regularity assumptions $\int w^\theta_{k|l} (dx_k) = 0$.

B. Recursive Maximum Likelihood

Recursive ML is a gradient type approach to maximize the average log-likelihood. This quantity has indeed "good" properties. Under suitable *regularity* conditions, see for example [22], one can show that

$$\frac{1}{n} l_\theta (Y^n) \to l(\theta)$$

with

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

where $\mathcal{P}(\mathbb{R}^p)$ is the space of probability distributions on $\mathbb{R}^p$ and $\lambda_{\theta,\theta^*}(dy,d\nu)$ is the joint invariant distribution of the couple $\left(Y_k, \mathcal{P}_\theta\left(X_k \in dx_k | Y^{k-1}\right)\right)$. It is dependent on both θ and the true parameter θ^*. The following Kullback–Leibler information measure satisfies

$$K(\theta,\theta^*) \triangleq l(\theta^*) - l(\theta) \geq 0 \tag{6}$$

and thus

$$\theta^* \subseteq \underset{\theta\in\Theta}{\arg\min}\; K(\theta,\theta^*).$$

To estimate θ^*, RML aims at minimizing $K(\theta,\theta^*)$ through a gradient method. The gradient of $K(\theta,\theta^*)$ does not admit an analytical expression but the gradient of $l_\theta(Y^n)$ satisfies, under regularity assumptions [22],

$$\frac{1}{n}(\partial l)_\theta(Y^n) \to -\partial K(\theta,\theta^*).$$

RML is a stochastic approximation algorithm based on the Robbins-Monro procedure to find the zeros of $-\partial K(\theta,\theta^*)$ [2], [19]:

$$\begin{aligned}
\theta_{n+1} &= \theta_n + \gamma_n \partial \log\left(\int \varepsilon_{\theta_n}(x_n,Y_n)\,\pi^{\theta^n}_{n|n-1}(dx_n)\right) &(7)\\
&= \theta_n + \gamma_n\left(\frac{\int(\partial\varepsilon)_{\theta_n}(x_n,Y_n)\,\pi^{\theta^n}_{n|n-1}(dx_n) + \int \varepsilon_{\theta_n}(x_n,Y_n)\,w^{\theta^n}_{n|n-1}(dx_n)}{\int \varepsilon_{\theta_n}(x_n,Y_n)\,\pi^{\theta^n}_{n|n-1}(dx_n)}\right) &(8)
\end{aligned}$$

where we denote $\pi^{\theta^n}_{n|n-1}(dx_n)$ the probability measure satisfying

$$\begin{aligned}
\pi^{\theta^n}_{n|n-1} &= \pi^{\theta^n}_{n-1|n-1} p_{\theta_n}, &(9)\\
\pi^{\theta^{n+1}}_{n|n} &= \frac{\pi^{\theta^n}_{n|n-1}.\varepsilon_{\theta_{n+1}}}{\left\langle \pi^{\theta^n}_{n|n-1}, \varepsilon_{\theta_{n+1}}\right\rangle}, &(10)
\end{aligned}$$

i.e. $\pi^{\theta^{n+1}}_{n|n}(dx_n)$ corresponds to the filter associated to the sequences of parameters values θ^{n+1}, the parameter being updated between the prediction step (see (9)) and the updating step (see (10)) using (7)-(8). The derivative of this measure $w^{\theta^n}_{n|n}(dx_n)$ is a signed measure satisfying

$$\begin{aligned}
w^{\theta^n}_{n|n-1} &= w^{\theta^n}_{n-1|n-1} p_{\theta_n} + \pi^{\theta^n}_{n-1|n-1}(\partial p)_{\theta_n}, &(11)\\
w^{\theta^{n+1}}_{n|n} &= \frac{w^{\theta^n}_{n|n-1}.\varepsilon_{\theta_{n+1}} + \pi^{\theta^n}_{n|n-1}.(\partial\varepsilon)_{\theta_{n+1}}}{\left\langle \pi^{\theta^n}_{n|n-1}, \varepsilon_{\theta_{n+1}}\right\rangle} - \pi^{\theta^{n+1}}_{n|n}\frac{\left\langle w^{\theta^n}_{n|n-1}, \varepsilon_{\theta_{n+1}}\right\rangle + \left\langle \pi^{\theta^n}_{n|n-1}, (\partial\varepsilon)_{\theta_{n+1}}\right\rangle}{\left\langle \pi^{\theta^n}_{n|n-1}, \varepsilon_{\theta_{n+1}}\right\rangle}. &(12)
\end{aligned}$$

The stepsize γ_n is a positive non-increasing sequence such that $\sum \gamma_n = \infty$ and $\sum \gamma_n^2 < \infty$; typically one selects $\gamma_n = \gamma_0 . n^{-\alpha}$ where $\gamma_0 > 0$ and $0.5 < \alpha \leq 1$ [2].

C. Convergence and Implementation Issues

In the context of finite state-space hidden Markov models, it is possible to compute the optimal filter and its derivative analytically. In this case, under weak assumptions, it can be established that (3) converges towards $\{\theta : (\partial l)_\theta (Y^n) = 0\}$ and (8) converges towards $\{\theta : \partial K(\theta, \theta^*) = 0\}$ [16]. Note that without any additional assumption these gradient methods are not ensured to converge to the true parameter value but only to the set of zeros of the gradient of the cost functions $l_\theta (Y^n)$ and $K(\theta, \theta^*)$ given in (6).

Using [22], similar results could also be established for general state-space models. However, in this case, the algorithms defined by (3) and (8) cannot be implemented as the filter and its derivative cannot be computed exactly and need to be approximated numerically. When these quantities are approximated, results in [21] suggest that (3) and (8) converge to a neighborhood of the zeros of the derivative of $l_\theta (Y^n)$ and $K(\theta, \theta^*)$.

Many efficient particle methods have been proposed to approximate the filter (see [5]) so we focus in this paper on approximating the derivative of the filter. We present in the next section an original particle method to address this problem. An alternative method has been proposed independently in [3] for a specific continuous-time model. The resampling step in their algorithm differs sensibly from ours.

III. Parameter Estimation using Particle Methods

In this section, we present a method to compute a particle approximation of $w^\theta_{n|n}$ in Subsection III-A. This method is based on an importance sampling resampling strategy. A few extensions of the algorithm are presented in Subsection III-B. Alternative approaches to estimate $w^\theta_{n|n}$ are then discussed in Subsection III-C.

A. Importance Sampling

For simplicity, we only describe here a particle method using the importance density $p_\theta(\cdot, \cdot)$ to approximate $\pi^\theta_{n|n}$ [8], [13]. More elaborate algorithms in [4], [5], [17] and [20] could also be used;

see Subsection III-B.

For two measures μ and ν on a common measurable space, we write $\mu \ll \nu$ if and only if $\nu(A) = 0 \Rightarrow \mu(A) = 0$ for all measurable sets A. Let us assume that $w^{\theta}_{n-1|n-1} \ll \pi^{\theta}_{n-1|n-1}$, then it can be shown using (11) and (12) that $w^{\theta}_{n|n-1} \ll \pi^{\theta}_{n|n-1}$ and $w^{\theta}_{n|n} \ll \pi^{\theta}_{n|n}$. This suggests that one can approximate $w^{\theta}_{n|n}$ by simply computing the importance ratio (i.e. Radon-Nykodym derivative) of $w^{\theta}_{n|n}$ with $\pi^{\theta}_{n|n}$. Let us assume that at time $n-1$, we have the following particle approximation of $\pi^{\theta}_{n-1|n-1}$

$$\widehat{\pi}^{\theta}_{n-1|n-1}(dx_{n-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\widehat{X}_{n-1,i}}(dx_{n-1}) \tag{13}$$

where $\delta_{\widehat{X}_{n-1,i}}(dx_{n-1})$ denotes the delta-Dirac measure at $\widehat{X}_{n-1,i}$ and

$$\widehat{w}^{\theta}_{n-1|n-1}(dx_{n-1}) = \sum_{i=1}^{N} \beta_{n-1|n-1,i}\delta_{\widehat{X}_{n-1,i}}(dx_{n-1}).$$

Note that in the above expression the coefficients $\beta_{n-1|n-1,i}$ can be negative or positive and do not sum to 1 as $w^{\theta}_{n-1|n-1}$ is not a probability measure. Using a particle interpretation of (11)-(12) and rewriting (11) as

$$w^{\theta}_{n|n-1} = w^{\theta}_{n-1|n-1}p_{\theta} + \pi^{\theta}_{n-1|n-1}p_{\theta}\frac{(\partial p)_{\theta}}{p_{\theta}},$$

one can derive a particle method to approximate $w^{\theta}_{n|n}$. It proceeds as follows at time n. We use the notation $\mathbb{I}_{A}(z) = 1$ if $z \in A$ and 0 otherwise.

Gradient Estimation using Particle Methods

Sampling step

- For $i = 1, ..., N$, sample $\widetilde{X}_{n,i} \sim p_{\theta}\left(\widehat{X}_{n-1,i}, \cdot\right)$.
- $\widetilde{\pi}^{\theta}_{n|n-1}(dx_n) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widetilde{X}_{n,i}}(dx_n)$ and $\widehat{w}^{\theta}_{n|n-1}(dx_n) = \sum_{i=1}^{N}\beta_{n|n-1,i}\delta_{\widetilde{X}_{n,i}}(dx_n)$, where

$$\beta_{n|n-1,i} = \beta_{n-1|n-1,i} + \frac{1}{N}\frac{(\partial p)_{\theta}\left(\widehat{X}_{n-1,i}, \widetilde{X}_{n,i}\right)}{p_{\theta}\left(\widehat{X}_{n-1,i}, \widetilde{X}_{n,i}\right)}.$$

Updating step

- $\widetilde{\pi}^{\theta}_{n|n}(dx_n) = \sum_{i=1}^{N}\widetilde{\alpha}_{n|n,i}\delta_{\widetilde{X}_{n,i}}(dx_n)$, where $\widetilde{\alpha}_{n|n,i} \propto \varepsilon_{\theta}\left(\widetilde{X}_{n,i}, Y_n\right)$, $\sum_{i=1}^{N}\widetilde{\alpha}_{n|n,i} = 1$.
- $\widetilde{w}^{\theta}_{n|n}(dx_n) = \sum_{i=1}^{N}\widetilde{\beta}_{n|n,i}\delta_{\widetilde{X}_{n,i}}(dx_n)$, where

$$\widetilde{\beta}_{n|n,i} = \frac{(\partial\varepsilon)_\theta\left(\widetilde{X}_{n,i},Y_n\right) + N\beta_{n|n-1,i}\varepsilon_\theta\left(\widetilde{X}_{n,i},Y_n\right)}{\sum_{j=1}^N \varepsilon_\theta\left(\widetilde{X}_{n,j},Y_n\right)} - \frac{\sum_{j=1}^N\left((\partial\varepsilon)_\theta\left(\widetilde{X}_{n,j},Y_n\right) + N\beta_{n|n-1,j}\varepsilon_\theta\left(\widetilde{X}_{n,j},Y_n\right)\right)}{\sum_{j=1}^N \varepsilon_\theta\left(\widetilde{X}_{n,j},Y_n\right)}\widetilde{\alpha}_{n|n,i}.$$

Resampling step

- Multiply/Discard particles $\widetilde{X}_{n,i}$ with respect to the high/low weights $\widetilde{\alpha}_{n|n,i}$ to obtain N particles $\widehat{X}_{n,i}$, i.e. $\widehat{X}_{n,i} = \widetilde{X}_{n,\varphi_n(i)}$ where $\varphi_n(i)$ is determined by the resampling mechanism.
- $\widehat{\pi}^\theta_{n|n}(dx_n) = \frac{1}{N}\sum_{i=1}^N \delta_{\widehat{X}_{n,i}}(dx_n)$ and $\widehat{w}^\theta_{n|n}(dx_n) = \sum_{i=1}^N \beta_{n|n,i}\delta_{\widehat{X}_{n,i}}(dx_n)$ where

$$\beta_{n|n,i} = \frac{\widetilde{\beta}^+_{n|n}}{\left(\widetilde{\beta}/\widetilde{\alpha}\right)^+_{n|n}}\frac{\widetilde{\beta}_{n|n,\varphi_n(i)}}{\widetilde{\alpha}_{n|n,\varphi_n(i)}}\mathbb{I}_{\mathbb{R}^+}\left(\widetilde{\beta}_{n|n,\varphi_n(i)}\right) + \frac{\widetilde{\beta}^-_{n|n}}{\left(\widetilde{\beta}/\widetilde{\alpha}\right)^-_{n|n}}\frac{\widetilde{\beta}_{n|n,\varphi_n(i)}}{\widetilde{\alpha}_{n|n,\varphi_n(i)}}\mathbb{I}_{\mathbb{R}^-}\left(\widetilde{\beta}_{n|n,\varphi_n(i)}\right),$$

with $\widetilde{\beta}^+_{n|n} \triangleq \sum_{i=1}^N \widetilde{\beta}_{n|n,i}\mathbb{I}_{\mathbb{R}^+}\left(\widetilde{\beta}_{n|n,i}\right)$, $\left(\widetilde{\beta}/\widetilde{\alpha}\right)^+_{n|n} \triangleq \sum_{i=1}^N \widetilde{\beta}_{n|n,\varphi_n(i)}/\widetilde{\alpha}_{n|n,\varphi_n(i)}\mathbb{I}_{\mathbb{R}^+}\left(\widetilde{\beta}_{n|n,\varphi_n(i)}\right)$, $\widetilde{\beta}^-_{n|n} \triangleq \sum_{i=1}^N \widetilde{\beta}_{n|n,i}\mathbb{I}_{\mathbb{R}^-}\left(\widetilde{\beta}_{n|n,i}\right)$, and $\left(\widetilde{\beta}/\widetilde{\alpha}\right)^-_{n|n} \triangleq \sum_{i=1}^N \widetilde{\beta}_{n|n,\varphi_n(i)}/\widetilde{\alpha}_{n|n,\varphi_n(i)}\mathbb{I}_{\mathbb{R}^-}\left(\widetilde{\beta}_{n|n,\varphi_n(i)}\right)$.

The computational complexity and memory requirements for this algorithm are in $\mathcal{O}(N)$ and independent of n.

Batch ML algorithm. At iteration m of the gradient algorithm, the particle method described above is runned from time 0 to T for the current parameter value θ_m. This allows to obtain $\widehat{\pi}^{\theta_m}_{k|k-1}$ and $\widehat{w}^{\theta_m}_{k|k-1}$. These quantities are used to obtain a Monte Carlo estimate of the gradient (5) using

$$\widehat{(\partial l)}_{\theta_m}(Y^n) = \sum_{k=1}^T \frac{\sum_{i=1}^N\left((\partial\varepsilon)_{\theta_m}\left(\widetilde{X}_{k,i},Y_k\right) + N\beta_{k|k-1,i}\varepsilon_{\theta_m}\left(\widetilde{X}_{k,i},Y_k\right)\right)}{\sum_{i=1}^N \varepsilon_{\theta_m}\left(\widetilde{X}_{k,i},Y_k\right)}.$$

This estimate is used to obtain θ_{m+1} using (3).

Recursive ML algorithm. At time n, we have obtained the sequence of parameters θ^n and $\widehat{\pi}^{\theta^n}_{n-1|n-1}$ and $\widehat{w}^{\theta^n}_{n-1|n-1}$. The sampling step of the particle method described above is runned with the parameter θ_n. After this step, one obtains $\widehat{\pi}^{\theta^n}_{n|n-1}$ and $\widehat{w}^{\theta^n}_{n|n-1}$. This allows to obtain θ_{n+1} using

$$\theta_{n+1} = \theta_n + \gamma_n\left(\frac{\sum_{i=1}^N\left((\partial\varepsilon)_{\theta_n}\left(\widetilde{X}_{n,i},Y_n\right) + N\beta_{n|n-1,i}\varepsilon_{\theta_n}\left(\widetilde{X}_{n,i},Y_n\right)\right)}{\sum_{i=1}^N \varepsilon_{\theta_n}\left(\widetilde{X}_{n,i},Y_n\right)}\right)$$

which is an approximation of (8). Then one runs the updating and resampling steps of the algorithm with the new parameter value to obtain $\widehat{\pi}^{\theta^{n+1}}_{n|n}$ and $\widehat{w}^{\theta^{n+1}}_{n|n}$.

Remark 1. *Resampling step.* The prediction and updating steps follow directly from a particle approximation of (11)-(12). The resampling step requires extra caution. The coefficients $\widetilde{\beta}_{n|n,\varphi_n(i)}$ are divided by $\widetilde{\alpha}_{n|n,\varphi_n(i)}$ so as to correct for the bias introduced by resampling particles according to $\widetilde{\alpha}_{n|n,i}$. It has also been designed so as to ensure that the masses of the positive and negative part of the signed measure $\widetilde{w}^{\theta^{n+1}}_{n|n}(dx_n)$ are preserved. Note that one could have also ensured the constraint $\sum_{i=1}^{N}\beta_{n|n,i}=0$ but this does not prove necessary. In the algorithm presented above, we resample the particles according to $\widetilde{\alpha}_{n|n,i}$. However, one can expect a high variance for this scheme if the zones of important masses for $\pi^{\theta^{n+1}}_{n|n}$ are distinct from that of $w^{\theta^{n+1}}_{n|n}$. It is possible to consider "hybrid" resampling schemes taking into account both $\widetilde{\alpha}_{n|n,i}$ and $\widetilde{\beta}_{n|n,i}$ so as to perform resampling. Optimizing this resampling scheme is the subject of further research.

Remark 3. *Reprojection.* Typically the parameter $\theta=(\theta_1,\ldots,\theta_k)$ belongs to Θ which is a compact convex subset of $\mathbb{R}^k$. The parameter updating step, see eq. (7), does not ensure that $\theta_{n+1}\in\Theta$ even if $\theta_n\in\Theta$. A standard approach in stochastic approximation to prevent divergence consists of reprojecting θ_{n+1} inside Θ whenever the value obtained through (7), say $\widetilde{\theta}_{n+1}$, does not belong to Θ [19]. Typically one has $\Theta=\prod_{i=1}^{k}[\theta_{i,\min},\theta_{i,\max}]$ and the reprojection procedure simply consists of setting $\theta_{i,n+1}=\theta_{i,\min}$ if $\widetilde{\theta}_{i,n+1}<\theta_{i,\min}$ and $\theta_{i,n+1}=\theta_{i,\max}$ if $\widetilde{\theta}_{i,n+1}>\theta_{i,\max}$.

B. Extensions

We discuss here several extensions of this algorithm.

Multidimensional parameter. If $\theta=(\theta_1,\ldots,\theta_k)$ where $k>1$, one needs to propagate k derivatives $\nabla\pi^{\theta}_{n|n-1}(dx_n)=\left(\partial_{\theta_1}\pi^{\theta}_{n|n-1}(dx_n),\ldots,\partial_{\theta_k}\pi^{\theta}_{n|n-1}(dx_n)\right)^{\mathrm{T}}=w^{\theta}_{n|n-1}(dx_n)$ then the updating step (8) of say the RML algorithm becomes

$$\begin{aligned}\theta_{n+1} &= \theta_n+\gamma_n\left(\int\varepsilon_{\theta_n}(x_n,Y_n)\,\pi^{\theta^n}_{n|n-1}(dx_n)\right)^{-1}\\ &\quad\times\left(\int(\nabla\varepsilon)_{\theta_n}(x_n,Y_n)\,\pi^{\theta^n}_{n|n-1}(dx_n)+\int\varepsilon_{\theta_n}(x_n,Y_n)\,w^{\theta^n}_{n|n-1}(dx_n)\right)\end{aligned}$$

where $(\nabla\varepsilon)_{\theta}(x_n,Y_n)=(\partial_{\theta_1}\varepsilon_{\theta}(x_n,Y_n),\ldots,\partial_{\theta_k}\varepsilon_{\theta}(x_n,Y_n))^{\mathrm{T}}$.

Alternative importance densities. Performance of particle filters can be improved significantly if one samples the particles according to an importance density $q_\theta(x_{n-1},Y_n,\cdot)$ (i.e. $\widetilde{X}_{n,i}\sim q_\theta\left(\widehat{X}_{n-1,i},Y_n,\cdot\right)$) different from the prior density $p_\theta(x_{n-1},\cdot)$ (with $p_\theta(x_{n-1},\cdot)\ll q_\theta(x_{n-1},Y_n,\cdot)$).

In this case, to perform parameter estimation, one uses

$$\begin{aligned}
&\log\left(\int \varepsilon_\theta\left(x_n,Y_n\right)\pi^\theta_{n|n-1}\left(dx_n\right)\right)\\
=\;&\log\left(\int\int \frac{\varepsilon_\theta\left(x_n,Y_n\right)p_\theta\left(x_{n-1},x_n\right)}{q_\theta\left(x_{n-1},Y_n,x_n\right)}q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\right),
\end{aligned}$$

where $\rho_\theta\left(x_{n-1},Y_n,x_n\right)=\frac{\varepsilon_\theta(x_n,Y_n)p_\theta(x_{n-1},x_n)}{q_\theta(x_{n-1},Y_n,x_n)}$ corresponds to the incremental unnormalized importance weight [4], [17]. It follows that in the parameter updating step of the algorithm (see eq. 7), one now uses

$$\begin{aligned}
&\partial\log\left(\int \varepsilon_\theta\left(x_n,Y_n\right)\pi^\theta_{n|n-1}\left(dx_n\right)\right)\\
=\;&\partial\log\left(\int\int \rho_\theta\left(x_{n-1},Y_n,x_n\right)q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\right)\\
=\;&\left(\int\int \rho_\theta\left(x_{n-1},Y_n,x_n\right)q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\right)^{-1}\\
&\times\left(\int\int\left(\partial\rho\right)_\theta\left(x_{n-1},Y_n,x_n\right)q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\right.\\
&\left.+\int\int \rho_\theta\left(x_{n-1},Y_n,x_n\right)\partial\left(q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\right)\right)
\end{aligned}$$

where

$$\begin{aligned}
\partial\left(q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\right) &= \left(\partial q\right)_\theta\left(x_{n-1},Y_n,x_n\right)dx_n\pi^\theta_{n-1|n-1}\left(dx_{n-1}\right)\\
&\quad+q_\theta\left(x_{n-1},Y_n,x_n\right)dx_n w^\theta_{n-1|n-1}\left(dx_{n-1}\right).
\end{aligned}$$

The particle method to approximate these equations follows straightforwardly.

Tracking a time-varying parameter. If the parameter θ is actually time-varying but one does not have a dynamic model for its evolution, a standard approach in control and signal processing to "track" this parameter consists of using the recursive algorithm presented before using a fixed-step size γ instead of a decreasing sequence γ_n. Bounds on the tracking errors can be established [2]. Selecting the step size is a difficult problem. If γ is too large, the statistical fluctuations around the parameter are too large. If γ is too small, the algorithm loses its tracking ability. In the context of adaptive filtering, adaptive step size schemes have been developed where the step size adapts itself automatically given the observations. It would be of interest to develop similar schemes in the context of general state-space models.

C. Alternative Approaches

1. Gradient Methods

We have investigated several alternative approaches to estimate $w^{\theta}_{n|n}$ using particle methods. The most natural one consists of using two distinct set of particles for $\pi^{\theta}_{n|n}$ and $w^{\theta}_{n|n}$. In this approach, one reinterprets (11) in a probabilistic way by using the fact that $(\partial p)_{\theta}$ can be rewritten as

$$(\partial p)_{\theta} = c_{p,\theta}\left(p'_{\theta} - p''_{\theta}\right) \tag{14}$$

where $c_{p,\theta} \in \mathbb{R}^{+}$, p'_{θ} and p''_{θ} are two probability transition kernels. There is actually an infinity of such decompositions; the most popular being the Hahn-Jordan decomposition where $p'_{\theta} \propto 0 \vee (\partial p)_{\theta}$ and $p''_{\theta} \propto -\left(0 \wedge (\partial p)_{\theta}\right)$. Another approach consists of rewriting

$$w^{\theta}_{n|n} = c_{n|n}\left(w'^{\theta}_{n|n} - w''^{\theta}_{n|n}\right)$$

where $c_{n|n} \in \mathbb{R}^{+}$, $w'^{\theta}_{n|n}$ and $w''^{\theta}_{n|n}$ are two probability measures. Using the fact that

$$(\partial \varepsilon)_{\theta} = c_{\varepsilon,\theta}\left(\varepsilon'_{\theta} - \varepsilon''_{\theta}\right)$$

where $c_{\varepsilon,\theta} \in \mathbb{R}^{+}$, ε'_{θ} and ε''_{θ} are two probability densities and the expressions (11)-(12)-(14), it is trivial to obtain recursions for $c_{n|n}$, $w'^{\theta}_{n|n}$ and $w''^{\theta}_{n|n}$. One can then propagate $w'^{\theta}_{n|n}$ and $w''^{\theta}_{n|n}$ using particle methods. Though these methods sound attractive, they are more computationally intensive than the importance sampling approach. Moreover, the algorithms we proposed to approximate these measures appear numerically unstable. This approach deserves however further study.

2. Non-gradient methods

As discussed briefly in Section I., several non-gradient approaches for parameter estimation using particle methods have been proposed in the literature. A simple approach consists of transforming fixed parameters into slowly time-varying ones [10], [14] or keeping the original model and combining the particle filter with MCMC steps [7]. The first method modifies the original problem whereas the second one suffers from an accumulation of errors over time and eventually diverges if the dataset is large. Another approach consists of discretizing the parameter space and evaluating at each point the likelihood [11]. This method is computationally expensive so one can only perform a coarse discretization of the parameter space. On the other hand, the main problem with gradient algorithms is their sensitivity to initial conditions. A pragmatic approach might consist of using one of the methods mentioned above to initialize a gradient method.

IV. Applications

In all examples, we apply a simple standard particle method using the prior density as importance density. As discussed in Subsection III-B, more elaborate algorithms could be developed using "tailored" importance densities. When Monte Carlo simulations are presented, new data have been simulated for each realization. We adopted $\gamma_n = \gamma_0 . n^{-0.8}$ for the stepsize sequence.

A. A Nonlinear Time Series

Let us consider the following model

$$\begin{aligned} X_{n+1} &= \cos\left(2\pi\phi X_n\right) + \sigma_v V_{n+1},\ X_0 \sim \mathcal{N}\left(0,2\right), \\ Y_n &= X_n + \sigma_w W_n \end{aligned}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0,1\right)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0,1\right)$ are two mutually independent sequences of independent identically distributed (i.i.d.) Gaussian random variables. We are interested in estimating the parameter $\theta \triangleq (\phi, \sigma_v, \sigma_w)$ where $\Theta = (0,1) \times (0,M) \times (0,M)$ where $M = 100$. Methods combining MCMC steps with particle filters cannot be used in this case as the conditional distribution of ϕ given the data and the hidden process is not in the exponential family.

The true parameter values are $\theta^* = (0.5, 1.0, 1.0)$. The RML algorithm was implemented using $N = 1000$ and $N = 10000$ particles and the initial parameter θ_0 was randomly selected. In the updating step of the algorithm, the parameter was reprojected inside Θ whenever necessary. In our 50 simulations, the algorithm appeared trapped in a local maximum located around $(0.71, 0.99, 1.10)$ in 12 simulations. For the remaining 38 simulations, the algorithm converges towards a value located around θ^*. In this case, as N increases, $\left\|\widehat{\theta} - \theta^*\right\|$ decreases; see Figure 1 for an example.

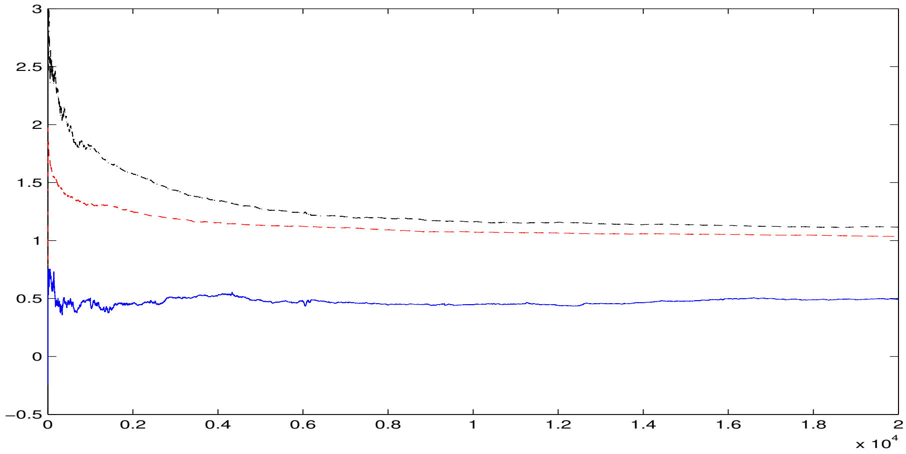

Figure 1: Sequence of parameter estimates $\theta_n = (\sigma_{vn}, \sigma_{wn}, \phi_n)$ for $N = 10000$. From top to bottom: σ_{vn}, σ_{wn} and ϕ_n.

B. Another Nonlinear Time Series

We consider the following nonlinear model [13], [14]

$$\begin{aligned} X_{n+1} &= \theta_3 X_n + \frac{\theta_4 X_n}{1 + X_n^2} + \theta_5 \cos(1.2(n+1)) + \theta_1 V_{n+1},\ X_0 \sim \mathcal{N}(0, 2), \\ Y_n &= \theta_6 X_n^2 + \theta_2 W_n \end{aligned}$$

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 independent sequences of i.i.d. Gaussian random variables. We are interested in estimating $\theta \triangleq (\theta_1, \ldots, \theta_6)$.

The true parameter values are $\theta^* = \left(1, \sqrt{10}, 0.5, 25, 8, 0.05\right)$. This problem is extremely complex. The likelihood function is highly multimodal. It is important in this example to initialize the algorithm properly. The batch ML algorithm was implemented using $N = 1000$ particles. With $\theta_0 = (0.5, 2, 1, 15, 4, 0.1)$, the algorithm converges towards to the neighborhood of the true parameter. We present in Figure 2 the sequence of parameter estimates (θ_4, θ_5). Clearly, there is a significant variance for the gradient estimate associated to θ_4.

C. Stochastic Volatility Model

Let us consider the following model [20]

$$\begin{aligned} X_{n+1} &= \phi X_n + \sigma V_{n+1},\ X_0 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right) \\ Y_n &= \beta \exp(X_n/2) W_n \end{aligned}$$

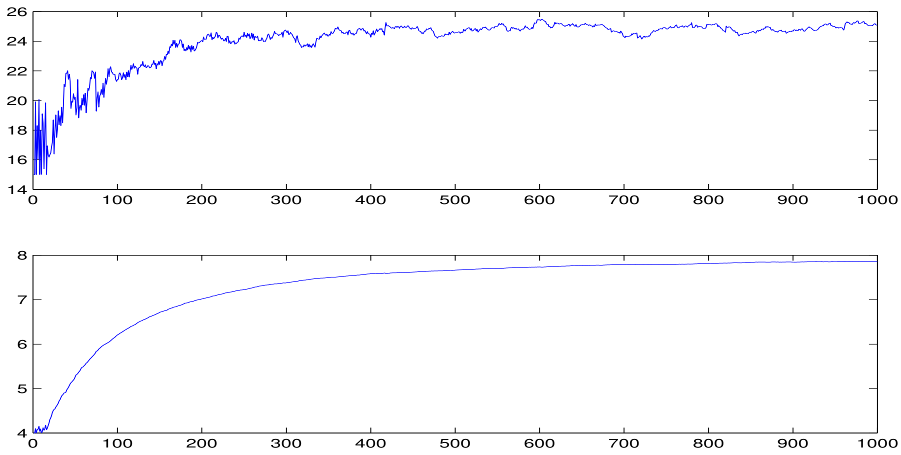

Figure 2: Sequence of parameter estimates for batch ML $\theta_{4,m}$ (top) and $\theta_{5,m}$ (bottom) for $N =$ 1000.

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, independent of the initial state X_0. We are interested in estimating the parameter $\theta \triangleq (\phi, \sigma, \beta)$ where $\Theta = (-1,1) \times \times (0, M) \times (0, M)$ where $M = 100$.

The true parameter values are $\theta^* = (0.8, 0.5, 1.0)$. We first implemented the RML algorithm using $N = 10000$ particles. As in the previous example, the parameter was reprojected inside Θ whenever necessary. In our 50 simulations, our algorithm did not appear sensitive to the initialization θ_0. It converged to a value $\widehat{\theta}$ in the neighborhood of θ^*; the larger N the smaller $\left\| \widehat{\theta} - \theta^* \right\|$; see Figure 3 for an example.

We then apply our batch ML method to the pound/dollar daily exchange rates; see [6] and [9]. This time series consists of 945 data. We apply the batch ML algorithm for $M = 1000$ iterations with $N = 10000$ particles; see Figure 4. Our results for the ML estimate are consistent with recent results obtained in [6]. We obtain $\widehat{\theta}_{ML} = (0.968, 0.188, 0.638)$ whereas the estimate in [6] is $\widehat{\theta}_{ML} = (0.973, 0.173, 0.634)$.

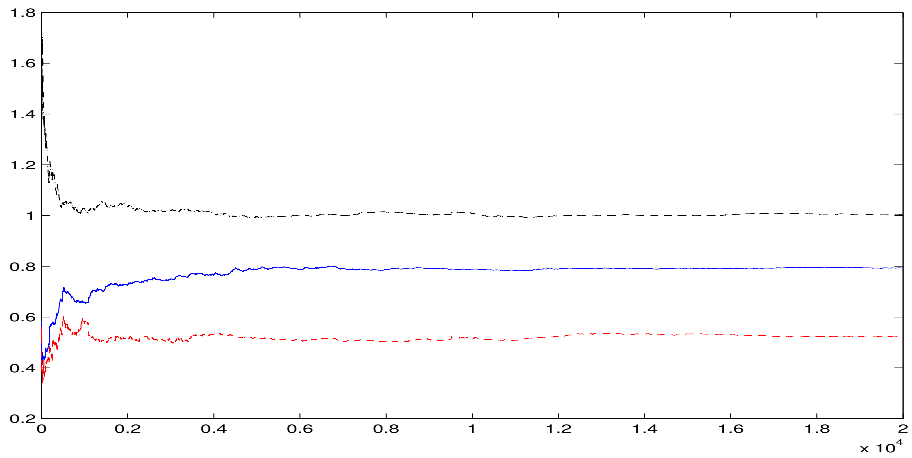

Figure 3: Sequence of parameter estimates for RML $\theta_n = (\beta_n, \phi_n, \sigma_n)$ for $N = 10000$. From top to bottom: β_n, ϕ_n and σ_n.

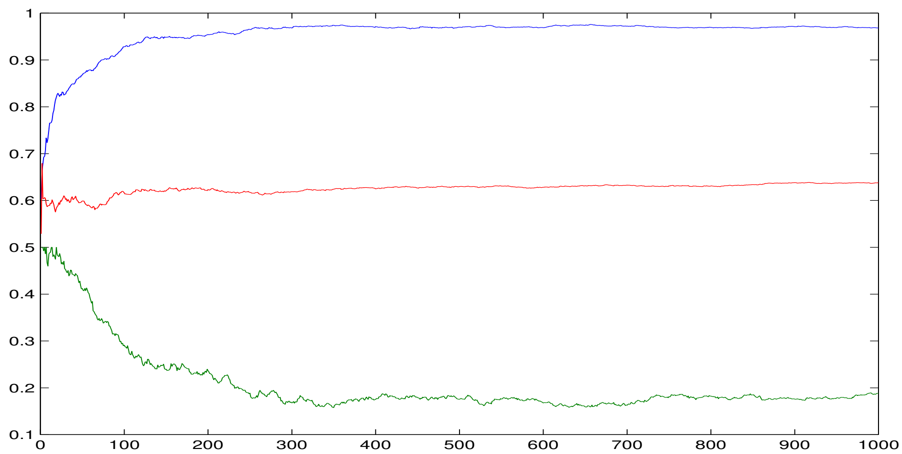

Figure 4: Sequence of parameter estimates for batch ML $\theta_m = (\beta_m, \phi_m, \sigma_m)$ for $N = 10000$. From top to bottom: β_m, ϕ_m and σ_m.

V. Discussion

In this paper, we have proposed two original gradient type algorithms to perform batch and recursive parameter estimation in general state-space models. These gradient algorithms require being able to estimate the derivatives of the filtering distribution with respect to the unknown parameters. We have proposed a simple method based on importance sampling resampling to estimate this quantity and have demonstrated the utility of our algorithms for nonlinear state-space models.

In the context of recursive parameter estimation, our algorithm requires a substantial amount of data to converge. While this is not a problem in most engineering applications, this could be a severe limitation in other contexts. To overcome this problem, it is necessary to develop alternative gradient estimation and variance reduction methods. Finally, from a theoretical viewpoint, non standard stochastic approximation results presented in [21] need to be used to prove convergence of the algorithms.

VI. Acknowledgments

The authors are grateful to the reviewers for their comments which allow to significantly improve this paper.

References

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

[2] Benveniste, A., Métivier, M. and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximation.* New York: Springer-Verlag.

[3] Cérou F., LeGland F. & 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 & A. Sulem), pp. 231-240, IOS Press, Amsterdam.

[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] Durbin J. and Koopman S. J. (2000) Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). *J. R. Statist. Soc.* B, **62**, pp. 3-56.

[7] 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.

[8] Gordon, N.J., Salmond, D.J. and Smith, A.F.M. (1993). Novel approach to nonlinear non-Gaussian Bayesian state estimation. *IEE Proc. F*, **140**, 107-113.

[9] Harvey, A.C., Ruiz, E. and Shephard, N. (1994) Multivariate stochastic variance models. *Rev. Econ. Stud.*, **61**, 247-264.

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

[11] Higuchi T. (2001) Evolutionary time series model with parallel computing. In *Proc. 3rd Japan-US Seminar on Statistical Time Series Analysis*, 183-190.

[12] Iba, Y. (2001). Population Monte Carlo algorithms. *Trans. Japanese Soc. Artificial Intelligence*, **16**, 279-286.

[13] Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, **5**, 1-25.

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

[15] Kitagawa. G. and Gersch, W. (1996). *Smoothness Priors Analysis of Time Series*, Lecture Notes in Statistics, vol. 116, New York: Springer.

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

[17] Liu, J.S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *J. Am. Statist. Ass.*, **93**, 1032-1043.

[18] 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.

[19] Ljung, L. and Söderström, T. (1987). *Theory and Practice of Recursive Identification*, MIT Press.

[20] Pitt, M.K. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filter. *J. Am. Statist. Ass.*, **94**, 590-599.

[21] Tadić, V.B. (2000). Asymptotic analysis of stochastic approximation algorithms under violated Kushner-Clark conditions. in *Proc. 39th IEEE Conf. Decision and Control.*

[22] Tadić, V.B. & Doucet A. (2002) Exponential forgetting and geometric ergodicity in state space models. in *Proc. 41th IEEE Conf. Decision and Control*, 2231-2235.