Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Gradient-free Maximum Likelihood Parameter Estimation with Particle Filters

George Poyiadjis, Sumeetpal S. Singh and Arnaud Doucet

Abstract— **In this paper we address the problem of on-line estimation of unknown static parameters in non-linear non-Gaussian state-space models. We consider a particle filtering method and employ two gradient-free Stochastic Approximation (SA) methods to maximize recursively the likelihood function, the Finite Difference SA and Spall's Simultaneous Perturbation SA. We demonstrate how these algorithms can generate maximum likelihood estimates in a simple and computationally efficient manner. The performance of the proposed algorithms is assessed through simulation.**

I. INTRODUCTION

Sequential Monte Carlo (SMC) methods, also known as particle filters, are a set of practical and flexible simulation-based techniques that have become increasingly popular to perform optimal filtering in non-linear non-Gaussian models [3], [4], [11]. However standard SMC methods assume knowledge of the model parameters. In many real-world applications, these parameters are unknown and need to be estimated. We address here the challenging problem of obtaining their Maximum Likelihood (ML) estimates. Previous approaches that extend the state with the unknown parameter and transform the problem into an optimal filtering problem - e.g. [5], [12], [21] - suffer from several drawbacks. Recently, a robust particle method to approximate the optimal filter derivative and perform maximum likelihood parameter estimation has been proposed [15]. This method is efficient but computationally intensive.

In general, gradient-based SA algorithms rely on a direct measurement of the gradient of an objective function with respect to the parameters of interest. Such an approach assumes that detailed knowledge of the system dynamics is available so that the gradient equations can be calculated. In the SMC framework, the gradient estimates of the particle approximations require a Likelihood Ratio or Infinitesimal Perturbation Analysis-based approach [16]. This usually results in a very high estimation variance that increases with the number of particles and with time. Although this problem can be successfully mitigated with a number of variance reduction techniques, this adds to the computational burden.

In this paper we investigate the use of gradient-free SA techniques as a simple alternative to generate ML parameters estimates. A related approach was used in [1] to optimize the performance of SMC algorithms. We adapt here this approach to parameter estimation. In principle, gradient-free techniques have a slower rate of convergence compared to gradient-based methods. However, gradient-free methods are only based on objective function measurements and do not require knowledge of the gradients of the underlying model. As a result, they are very easy to implement and have a reduced computational complexity. The classical gradient-free method is the Kiefer-Wolfworitz Finite Difference SA (FDSA) algorithm [7]. A more efficient approach that has recently attracted attention is the Simultaneous Perturbation SA (SPSA) method introduced by Spall [17]. This is based on a randomized finite different method and it is particularly attractive in high dimensional optimization problems. Both methods are considered here.

The remainder of the paper is organized as follows: Section II describes the optimal filtering problem and the SMC framework. In Section III we formalize the parameter estimation problem and outline our solution methodology. In Section IV we describe the proposed gradient-free algorithms. Some applications demonstrating the efficiency of the methods are presented in Section V. Finally in Section VI we discuss the results and provide some concluding remarks.

II. OPTIMAL FILTERING USING SMC METHODS

A. State-Space Models

Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 0}$ be $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$ -valued stochastic processes defined on a measurable space $(\Omega, \mathcal{F})$. Let $\theta \in \Theta$ be the parameter vector where Θ is an open subset of $\mathbb{R}^m$. A general discrete-time state-space model represents the unobserved state $\{X_n\}_{n\geq 0}$ as a Markov process of initial density $X_0 \sim \mu$ and Markov transition density $f_\theta(x'|x)$. The observations $\{Y_n\}_{n\geq 0}$ are assumed conditionally independent given $\{X_n\}_{n\geq 0}$ and are characterized by their conditional marginal density $g_\theta(y|x)$. The model is summarized as follows

$$X_n|X_{n-1} = x_{n-1} \sim f_\theta(\,.\,|x_{n-1}), \tag{1}$$

$$Y_n|X_n = x_n \sim g_\theta(\,.\,|x_n), \tag{2}$$

where the two densities can be non-Gaussian and may involve non-linearities. For any sequence $\{z_k\}$ and random process $\{Z_k\}$ we will use the notation $z_{i:j} = (z_i, z_{i+1}, ..., z_j)$ and $Z_{i:j} = (Z_i, Z_{i+1}, ..., Z_j)$.

Assume for the time being that θ is known. In such a situation, one is interested in estimating the hidden state X_n given the observation sequence $\{Y_n\}_{n\geq 0}$. This leads to the so-called optimal filtering problem that seeks to compute the posterior density $p_\theta(x_n|Y_{0:n})$ sequentially in time. Introducing a proposal distribution $q_\theta\left(x_n|\,Y_n, x_{n-1}\right)$, whose

G. Poyiadjis and S.S. Singh are with the Signal Processing and Communications Group, Department of Engineering, University of Cambridge, CB2 1PZ, UK {gp243,sss40@cam.ac.uk}

A. Doucet is with the Department of Computer Science and the Department of Statistics, University of British Columbia, Vancouver, BC, Canada. arnaud@stat.ubc.ca

1-4244-0210-7/06/$20.00 ©2006 IEEE

support includes the support of $g_\theta\left(Y_n|\,x_n\right) f_\theta\left(x_n|\,x_{n-1}\right)$, the filtering density satisfies the recursion

$$\begin{aligned} p_\theta\left(x_n|\,Y_{0:n}\right) \propto \int & \alpha_\theta\left(x_{n-1:n}, Y_n\right) \\ & \times\, q_\theta\left(x_n|\,Y_n, x_{n-1}\right) p_\theta\left(x_{n-1}|\,Y_{0:n-1}\right) dx_{n-1} \end{aligned} \tag{3}$$

where $\alpha_\theta\left(x_{n-1:n}, Y_n\right) = \dfrac{g_\theta\left(Y_n|\,x_n\right) f_\theta\left(x_n|\,x_{n-1}\right)}{q_\theta\left(x_n|\,Y_n, x_{n-1}\right)}$. (4)

Except in some very simple cases, no closed-form expression can be obtained for this recursion and numerical approximations are required.

B. SMC framework

SMC methods approximate the optimal filtering density by a weighted empirical distribution; i.e. a weighted sum of $N \gg 1$ samples, termed as particles. Here we will assume that at time $n-1$, the filtering density $p_\theta\left(x_{n-1}|Y_{0:n-1}\right)$ is approximated by the particle set $X_{n-1}^{(1:N)} \triangleq \left[X_{n-1}^{(1)}, \ldots, X_{n-1}^{(N)}\right]$ having equal weights. The filtering distribution at the next time step can be recursively approximated by a new set of particles $X_n^{(1:N)}$ generated via an importance sampling and a resampling step.

In the importance sampling step, a set of prediction particles are generated independently from $\widetilde{X}_n^{(i)} \sim q_\theta\left(\cdot|\; Y_n, X_{n-1}^{(i)}\right)$ and are weighted by an importance weight $\widetilde{a}_{\theta,n}^{(i)}$ that accounts for the discrepancy with the "target" distribution. This is given by

$$a_{\theta,n}^{(i)} = \alpha_\theta\left(\widetilde{X}_n^{(i)}, X_{n-1}^{(i)}, Y_n\right) \text{ and} \tag{5}$$

$$\widetilde{a}_{\theta,n}^{(i)} = \frac{a_{\theta,n}^{(i)}}{\sum_{j=1}^N a_{\theta,n}^{(j)}}. \tag{6}$$

In the resampling step, the particles $\widetilde{X}_n^{(1:N)}$ are multiplied or eliminated according to their importance weights $\widetilde{a}_{\theta,n}^{(1:N)}$ to give the new set of particles $X_n^{(1:N)}$, based on the mapping

$$\begin{aligned} X_n^{(1:N)} &= H\left(\widetilde{X}_n^{(1:N)}, I_n^{(1:N)}\right) \\ &\triangleq [\underbrace{\widetilde{X}_n^{(1)}, \ldots, \widetilde{X}_n^{(1)}}_{I_n^{(1)} \text{ times}}, \ldots, \underbrace{\widetilde{X}_n^{(N)}, \ldots, \widetilde{X}_n^{(N)}}_{I_n^{(N)} \text{ times}}] \end{aligned} \tag{7}$$

where $I_n^{(i)}$ represents the number of copies of particle $\widetilde{X}_n^{(i)}$. The resampling index vector $I_n^{(1:N)} \triangleq \left[I_n^{(1)}, \ldots, I_n^{(N)}\right]$ can be obtained using standard methods such as multinomial, residual or systematic resampling. The full algorithm is summarized as follows:

Generic Sequential Monte Carlo algorithm (SIR)

At time $n-1$, assume that a set of equally weighted particles $X_{n-1}^{(1:N)} = \left[X_{n-1}^{(1)}, \ldots, X_{n-1}^{(N)}\right]$ is available.

Importance sampling step

- For $i = 1, ..., N$, sample $\widetilde{X}_n^{(i)} \sim q_\theta\left(\cdot \mid Y_n, X_{n-1}^{(i)}\right)$ and evaluate the weights $\widetilde{a}_{\theta,n}^{(i)}$ using (5), (6).

Weighted resampling step

- Sample $I_n^{(1:N)} \sim L\left(\cdot \mid \widetilde{a}_{\theta,n}^{(1:N)}\right)$ using a standard resampling scheme.
- Set $X_n^{(1:N)} = H\left(\widetilde{X}_n^{(1:N)}, I_n^{(1:N)}\right)$.

Note that the standard bootstrap filter [6] corresponds to the case where $q_\theta\left(\widetilde{X}_n^{(i)} \mid Y_n, X_{n-1}^{(i)}\right) = p_\theta\left(\widetilde{X}_n^{(i)} \mid X_{n-1}^{(i)}\right)$ and the distribution for $I_n^{(1:N)}$ is a multinomial distribution of parameters $\widetilde{a}_{\theta,n}^{(1:N)}$.

III. Problem Statement and Solution Methodology

Let us now consider the case where the model includes some unknown parameters. We will assume that the system to be identified evolves according to a true but unknown static parameter θ^*, i.e.

$$X_n|X_{n-1} = x_{n-1} \sim f_{\theta^*}(\,.\,|x_{n-1}) \tag{8}$$

$$Y_n|X_n = x_n \sim g_{\theta^*}(\,.\,|x_n). \tag{9}$$

The aim is to identify this parameter. Addressing this problem for a non-Gaussian and non-linear system is very challenging.

We aim to identify θ^* based on an infinite (or very large) observation sequence $\{Y_n\}_{n\geq 0}$, in an on-line fashion. A standard method to do so is to maximize the limit of the time averaged log-likelihood function:

$$l\left(\theta\right) = \lim_{k\to\infty} \frac{1}{k+1} \sum_{n=0}^{k} \log p_\theta(Y_n\,|Y_{0:n-1}\,) \tag{10}$$

with respect to θ. Suitable regularity conditions ensure that this limit exist and $l\left(\theta\right)$ admits θ^* as a global maximum [22]. The expression $p_\theta(Y_n\,|Y_{0:n-1}\,)$ is the predictive likelihood and can be written as

$$\begin{aligned} p_\theta(Y_n\,|Y_{0:n-1}\,) = \int\int & \alpha_\theta\left(x_{n-1:n}, Y_n\right) q_\theta\left(x_n|\,Y_n, x_{n-1}\right) \\ & \times p_\theta\left(x_{n-1}|\,Y_{0:n-1}\right) dx_{n-1:n}. \end{aligned} \tag{11}$$

Note that this is the normalization constant of (3). This approach is known as recursive maximum likelihood parameter estimation [13].

Our contribution is a novel recursive ML parameter estimation scheme based on the SMC algorithm described in the previous section. Unfortunately it is impossible to compute $\log p_\theta(Y_n\,|Y_{0:n-1}\,)$ in closed form. Instead, we use a particle approximation and propose to optimize an alternative criterion: the SMC algorithm provides us with samples $\left(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ from $p_\theta\left(x_{n-1}|\,Y_{0:n-1}\right) q_\theta\left(x_n|\,Y_n, x_{n-1}\right)$. Therefore a particle approximation to $\log p_\theta(Y_n\,|Y_{0:n-1}\,)$ is given by

$$\log \widehat{p_\theta}\left(Y_n\,|Y_{0:n-1}\,\right) = \log\left(N^{-1}\sum_{i=1}^{N} a_{\theta,n}^{(i)}\right). \tag{12}$$

Now we use the key fact that the current hidden state X_n, the observation Y_n, the predicted particles $\widetilde{X}_n^{(1:N)}$ and their corresponding unnormalized weights $a_{\theta,n}^{(1:N)}$ form a homogeneous Markov chain. We will denote this Markov chain by $\{Z_n\}_{n\geq 0}$, where:

$$Z_n = \left(X_n, Y_n, \widetilde{X}_n^{(1:N)}, a_{\theta,n}^{(1:N)}\right).$$

Let us also define a "reward" function $r(Z_n)$ as follows,

$$r\left(X_n, Y_n, \widetilde{X}_n^{(1:N)}, a_{\theta,n}^{(1:N)}\right) = \log\left(N^{-1}\sum_{i=1}^{N} a_{\theta,n}^{(i)}\right).$$

Note that $r(Z_n)$ is precisely $\log \widehat{p_\theta}\left(Y_n \left| Y_{0:n-1}\right.\right)$ in (12). The new criterion we seek to maximize is the particle approximation to the time averaged log-likelihood of (10) that is given by

$$J\left(\theta\right) = \lim_{k\to\infty}\frac{1}{k+1}\sum_{n=0}^{k}\log r\left(Z_n\right).$$

If the Markov chain $\{Z_n\}_{n\geq 0}$ is ergodic and admits an invariant distribution $\pi_{\theta,\theta^*}\left(\cdot\right)$, then this limit exists and is equal to

$$J\left(\theta\right) = E_{\pi_{\theta,\theta^*}}\left[r\left(Z\right)\right],$$

where Z is distributed according to π_{θ,θ^*}. This is true irrespective of the initial distribution of the state of the chain [14]. Note that the invariant distribution $\pi_{\theta,\theta^*}\left(\cdot\right)$ is a function of both θ^* and θ. This is because the first two components of Z_n, i.e. (X_n, Y_n), evolve according to the true parameter θ^* that we wish to identify. On the other hand, the particle filter components of Z_n, i.e. $\left(\widetilde{X}_n^{(1:N)}, a_{\theta,n}^{(1:N)}\right)$, evolve according to θ.

In the following section, we propose SA algorithms to solve

$$\vartheta^* = \arg\max_{\theta\in\Theta} J\left(\theta\right).$$

Note that because we only use a finite number N of particles, $\left(\widetilde{X}_n^{(1:N)}, a_{\theta,n}^{(1:N)}\right)$ is only an approximation to the exact prediction density $p_\theta\left(x_n | Y_{0:n-1}\right)$. Hence ϑ^* will not be equal to the true parameter θ^*. However, as N increases, $J\left(\theta\right)$ will get closer to $l\left(\theta\right)$ and ϑ^* will converge to θ^*. Our simulation results indicate that ϑ^* provides a good approximation to θ^* for a moderate number of particles.

IV. Maximum Likelihood Estimation using Gradient-free SA

We are interested in maximizing $J\left(\theta\right)$ with respect to the m-dimensional parameter vector θ. The function $J\left(\theta\right)$ does not admit an analytical expression. Additionally, we do not have access to it. Using the geometric ergodicity of the Markov chain $\{Z_n\}_{n\geq 0}$, $J\left(\theta\right)$ can be approximated in the limit as follows,

$$J\left(\theta\right) = \lim_{n\to\infty}\left\{J_n\left(\theta\right) \triangleq E_{\theta,\theta^*}\left[r\left(Z_n\right)\right]\right\}, \quad (13)$$

where the expectation is taken with respect to the distribution of Z_n. This implies that although $J\left(\theta\right)$ is unknown, we have access to a sequence of functions J_n that converge to $J\left(\theta\right)$. One way to exploit this sequence in order to optimize $J(\theta)$, is to use a recursion as follows,

$$\theta_n = \theta_{n-1} + \gamma_n \widehat{\nabla J}_n\left(\theta_{n-1}\right) \quad (14)$$

where θ_{n-1} is the parameter estimate at time $n-1$ and $\widehat{\nabla J}_n$ denotes an estimate of ∇J_n (preferably unbiased) [1]. The idea is that we take incremental steps to improve θ where each step uses a particular function from the sequence. Under suitable conditions on the step size, the above iteration will converge to ϑ^* [10].

We will consider the case where the expression for the gradient of J_n is either not available or too complex to calculate. Note that analytic expressions for the model in (8)-(9) are not always available and without which, the gradient of J_n cannot be derived. One may approximate $\widehat{\nabla J}_n\left(\theta\right)$ by recourse to finite difference methods. These are "gradient-free" methods that only use measurements of $J_n\left(\theta\right)$. The idea behind this approach is to measure the change in the function induced by a small perturbation $\Delta\theta$ in the value of the parameter. If we denote an estimate of $J_n\left(\theta\right)$ by $\widehat{J_n}\left(\theta\right)$[2], one-sided gradient approximations consider the change between $\widehat{J_n}\left(\theta\right)$ and $\widehat{J_n}\left(\theta+\Delta\theta\right)$, while two-sided approximations consider the difference between $\widehat{J_n}\left(\theta-\Delta\theta\right)$ and $\widehat{J_n}\left(\theta+\Delta\theta\right)$. In cases where it is possible to obtain samples from (1) for any θ but the functional form $f_\theta\left(\cdot|\cdot\right)$ is unknown, then a gradient-free approach is the only possibility. On the other hand, even if a gradient can be derived, it is often the case that gradient calculations are very involved. A gradient-free approach can provide a maximum likelihood parameter estimate that is computationally cheap, as well as very simple to implement. In principle however, if direct measurements of the gradient can be computed with reasonable effort, gradient-based methods should be used.

A. Finite Difference Stochastic Approximation

The classical method for gradient-free optimization is the Kiefer-Wolfwitz FDSA technique introduced in 1952 [7]. In this approach, each component of θ is perturbed one at a time and the corresponding changes in the objective function are used to approximate the gradient. In the two-sided case, the μ^{th} component of the gradient estimate $\widehat{\nabla J}_n\left(\theta\right) = \left[\widehat{\nabla J}_{n,1}\left(\theta\right), \ldots, \widehat{\nabla J}_{n,m}\left(\theta\right)\right]$ is given by

$$\widehat{\nabla J}_{n,\mu}\left(\theta\right) = \frac{\widehat{J_n}\left(\theta + c_n e_\mu\right) - \widehat{J_n}\left(\theta - c_n e_\mu\right)}{2c_n},$$

where e_μ denotes a row vector of size m with the value '1' in the μ^{th} entry and '0' elsewhere and $\{c_n\}_{n\geqslant 1}$ is a sequence of small positive numbers that typically get smaller with time n. Exact conditions on the choice of the step sizes are given below. The motivation behind the FDSA approach can be easily understood by observing that in the limiting case it leads to the standard definition of the gradient as a vector of partial derivatives, i.e. $\nabla_\theta J\left(\theta\right) = \left[\frac{\partial J(\theta)}{\partial\theta_1}, \ldots, \frac{\partial J(\theta)}{\partial\theta_m}\right]$. Note that for each gradient estimate, the two-sided FDSA method requires $2m$ cost function evaluations, where m is the dimension of the gradient vector. Similarly, the one-sided version would need $m+1$ evaluations. In scenarios where m is large, it is therefore judicious to consider more efficient methods such as the SPSA technique.

[1] For a real valued function $y = f\left(\mathbf{x}\right)$ of the row vector $\mathbf{x} = [x_1, \ldots, x_n]$, the operation $\nabla_{\mathbf{x}} f\left(\mathbf{x}\right)$ will denote the row vector of the partial derivatives of $f\left(\mathbf{x}\right)$, i.e. $\nabla_{\mathbf{x}} f\left(\mathbf{x}\right) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right]$.

[2] In order to obtain an unbiased estimate of $J_n\left(\theta\right)$ we may simulate the Markov Chain until time n, with a fixed value of θ, and use $r(Z_n)$. This is a standard approach [14] that we adopt in this paper. See also Remark 1.

B. Simultaneous Perturbation Stochastic Approximation

The key feature of the SPSA technique is that it requires only two measurements of the cost function regardless of the dimension of the parameter vector. This efficiency is achieved by the fact that all the elements in θ are perturbed together. The μ^{th} component of the two-sided gradient approximation $\widehat{\nabla J}_n(\theta) = \left[\widehat{\nabla J}_{n,1}(\theta), \ldots, \widehat{\nabla J}_{n,m}(\theta)\right]$ is given by

$$\widehat{\nabla J}_{n,\mu}(\theta_{n-1}) = \frac{\widehat{J_n}(\theta_{n-1} + c_n\Delta_n) - \widehat{J_n}(\theta_{n-1} - c_n\Delta_n)}{2c_n\Delta_{n,\mu}}$$

where $\Delta_n \triangleq [\Delta_{n,1}, \ldots, \Delta_{n,m}]$ is a random perturbation vector and $\{c_n\}_{n\geqslant 1}$ is defined as before. Note that the computational saving stems from the fact that the objective function difference is now common in all m components of the gradient approximation vector.

Almost sure convergence of the SA recursion in (14) is guaranteed if $J(\theta)$ is sufficiently smooth (several times differentiable) near ϑ^*. Additionally, the elements of Δ_n must be mutually independent random variables, symmetrically distributed around zero and with finite inverse moments $\mathbb{E}\left(|\Delta_{n,\mu}|^{-1}\right)$. A simple and popular choice for Δ_n that satisfies these requirements is the Bernoulli ± 1 distribution[3]. Furthermore, the positive step sizes should satisfy

$$\gamma_n \longrightarrow 0,\ c_n \longrightarrow 0,\ \sum_{n=1}^{\infty} \gamma_n = \infty \text{ and} \sum_{n=1}^{\infty} \left(\frac{\gamma_n}{c_n}\right)^2 < \infty.$$

See [17] for more details on convergence conditions.

The choice of the step sequences is crucial to the performance of the algorithm. Some guidelines on the general form of the step sequences and some typical choices that have been found effective for particular applications are discussed in [18] and [19]. Note that if a constant step size is used for γ_n, the SA estimate will still converge but will oscillate about the limiting value with a variance proportional to the step size. In most of our simulations γ_n was set to a small constant step size that was repeatedly halved after several thousands of iterations. For c_n we used a decreasing sequence of the form $c_n = \frac{c}{n^\tau}$, where c and τ are non-negative coefficients.

C. Convergence Acceleration

A number of convergence acceleration methods can be used to improve the performance of the algorithms. These are discussed below.

a) Common Random numbers: In gradient-free SA methods, the gradient approximation is the difference of two objective function measurements, each based on a different realization of the same system. For the two-sided SPSA case for example, these would be $\widehat{J_n}(\theta + c_n\Delta_n; \omega_n^+)$ and $\widehat{J_n}(\theta - c_n\Delta; \omega_n^-)$, where ω_n^+ and ω_n^- denote the randomness of each realization. This implies that besides the desired objective function change induced by the perturbation in θ, there is also some undesirable variability in $\widehat{\nabla J}_n(\theta)$ due to the underlying randomness $\omega_n^{\pm}$. Although in a real system $\omega_n^{\pm}$ cannot be controlled, in simulation settings it might be possible to eliminate the undesirable variability component by using the same random seeds at every time instant n, so that $\omega_n^+ = \omega_n^-$. This common random numbers idea leads to faster convergence of the algorithm [8]. Common random numbers were used in all our simulations.

b) Adaptive steps: The SA of (14) can be thought of as a stochastic generalization of the steepest descent method. Faster convergence can be achieved if one uses a Newton-type SA algorithm that is based on an estimate of the second derivative of the objective function. This will be of the form

$$\theta_n = \theta_{n-1} - \gamma_n \left[\widehat{\nabla^2 J}_n(\theta_{n-1})\right]^{-1} \widehat{\nabla J}_n(\theta_{n-1}), \qquad (15)$$

where $\widehat{\nabla^2 J_n}$ is an estimate of the negative definite Hessian matrix $\nabla^2 J_n$. Such an approach is particularly attractive in terms of convergence acceleration, in the terminal phase of the algorithm, where the steepest descent-type method of (14) slows down. The main difficulty with second order methods is the fact that the estimate of the Hessian should also be a negative definite matrix. This is usually ensured by projecting the Hessian estimate onto the set of negative definite matrices, before using it in (15); see [2].

Finite difference methods are usually the only practical choice for the Hessian estimate since extensions of gradient-based methods to second order derivatives are in general non-trivial. A simultaneous perturbation-based second order algorithm has been proposed in [20]. The method estimates the Hessian using a recursion that runs in parallel with the SA of (15) and computes the running average of the Hessian estimates. The algorithm requires only a small number of cost function evaluations that, as in the standard SPSA algorithm, are independent of the dimensions of the parameter vector.

c) Perturbation averaging: As it was suggested in [17], it might be useful to average several simultaneous perturbation gradient approximations at each iteration, each with an independent value of Δ_n. Despite the expense of additional objective function evaluations, this can reduce the noise effects and accelerate convergence.

D. Parameter Estimation using FDSA and SPSA

In this section we present two maximum likelihood parameter estimation algorithms within the SMC framework that are based on a two-sided FDSA and a two-sided SPSA method. In line with our objectives, the algorithm below only requires a single realization of observations $\{Y_n\}_{n\geq 1}$ of the true system (8)-(9). Furthermore, the algorithm operates in an on-line fashion and does not need to revisit the past observations.

At time $n-1$, we denote the current parameter estimate by θ_{n-1}. Also, let the filtering density $p_{\theta_{0:n-1}}(x_{n-1}|Y_{0:n-1})$ be approximated by the particle set $X_{n-1}^{(1:N)}$ having equal importance weights. Note that the subscript $\theta_{0:n-1}$ indicates that the filtering density estimate is a function of all the past parameter values. The FDSA/SPSA algorithms at time n proceed as follows:

[3] Note that the uniform and the normal distributions do not satisfy the finite inverse moment condition and are therefore unsuitable.

```
Recursive Parameter Estimation using FDSA/SPSA
(Execute option 1 for FDSA or option 2 for SPSA)

Option 1: Gradient approximation using FDSA
• For each parameter component $\mu = 1, ..., m$ and
    - for each particle $i = 1, ..., N$, sample
        $\widetilde{X}^{(i)}_{n,\mu^+} \sim q_{\theta_{n-1}+c_n e_\mu}\left(\cdot \mid Y_n, X^{(i)}_{n-1}\right)$,
        $\widetilde{X}^{(i)}_{n,\mu^-} \sim q_{\theta_{n-1}-c_n e_\mu}\left(\cdot \mid Y_n, X^{(i)}_{n-1}\right)$
        and using (4) evaluate
        $a_\theta\left(Y_n, \widetilde{X}^{(i)}_{n,\mu^+}, X^{(i)}_{n-1}\right)$, $a_\theta\left(Y_n, \widetilde{X}^{(i)}_{n,\mu^-}, X^{(i)}_{n-1}\right)$.
• Evaluate $\widehat{\nabla J}_{n,\mu}\left(\theta_{n-1}\right) = \frac{\widehat{J_n}(\theta_{n-1}+c_n e_\mu)-\widehat{J_n}(\theta_{n-1}-c_n e_\mu)}{2c_n}$
  where $\widehat{J_n}\left(\theta_{n-1} \pm c_n e_\mu\right) =$
  $\log\left\{\frac{1}{N}\sum_{i=1}^{N} a_{\theta_{n-1}\pm c_n e_\mu}\left(Y_n, \widetilde{X}^{(1:N)}_{n,\mu^\pm}, X^{(i)}_{n-1}\right)\right\}$.

Option 2: Gradient approximation using SPSA
• Generate random perturbation vector $\Delta_n$.
• For $i = 1, ..., N$, sample
    $\widetilde{X}^{(i)}_{n,+} \sim q_{\theta_{n-1}+c_n \Delta_n}\left(\cdot \mid Y_n, X^{(i)}_{n-1}\right)$,
    $\widetilde{X}^{(i)}_{n,-} \sim q_{\theta_{n-1}-c_n \Delta_n}\left(\cdot \mid Y_n, X^{(i)}_{n-1}\right)$
  and using (4) evaluate
  $a_\theta\left(Y_n, \widetilde{X}^{(i)}_{n,+}, X^{(i)}_{n-1}\right)$, $a_\theta\left(Y_n, \widetilde{X}^{(i)}_{n,-}, X^{(i)}_{n-1}\right)$.
• Evaluate $\widehat{\nabla J}_{n,\mu}\left(\theta_{n-1}\right)=\frac{\widehat{J_n}(\theta_{n-1}+c_n \Delta_n)-\widehat{J_n}(\theta_{n-1}-c_n \Delta_n)}{2c_n \Delta_{n,\mu}}$
  where $\widehat{J_n}\left(\theta_{n-1} \pm c_n \Delta_n\right) =$
  $\log\left\{\frac{1}{N}\sum_{i=1}^{N} a_{\theta_{n-1}\pm c_n \Delta_n}\left(Y_n, \widetilde{X}^{(1:N)}_{n,\pm}, X^{(i)}_{n-1}\right)\right\}$.

Parameter update step
• $\theta_n = \theta_{n-1} + \gamma_n \widehat{\nabla J}_n\left(\theta_{n-1}\right)$, where
  $\widehat{\nabla J}_n\left(\theta_{n-1}\right)\left[\widehat{\nabla J}_{n,1}\left(\theta_{n-1}\right), \ldots, \widehat{\nabla J}_{n,m}\left(\theta_{n-1}\right)\right]$.

Particle Filter
• For $i = 1, ..., N$, sample $\widetilde{X}^{(i)}_n \sim q_{\theta_n}\left(\cdot \mid Y_n, X^{(i)}_{n-1}\right)$
  and evaluate the weights $\widetilde{a}^{(i)}_{\theta_n,n}$ using (5), (6).
• Sample $I^{(1:N)}_n \sim L\left(\cdot \mid \widetilde{a}^{(1:N)}_{\theta,n}\right)$ using a standard
  resampling scheme.
• Set $X^{(1:N)}_n = H\left(\widetilde{X}^{(1:N)}_n, I^{(1:N)}_n\right)$.
```

Remark 1: For a Markov Chain $\{Z_n\}_{n\geq 0}$with a fixed initial distribution and a transition density parameterized by θ, an unbiased estimate of $E_{\theta,\theta^*}\left[r\left(Z_n\right)\right]$is obtained by simulating the chain until time n, while holding θfixed, and using $r\left(Z_n\right)$. In our problem, $\theta$is the parameter we are estimating recursively and will not be fixed. However, since θchanges slowly, a standard approach is to reuse the trajectories $Z_{0:n-1}$that were simulated with $\theta_{0:n-1}$and still use $r\left(Z_n\right)$as the estimate [14].

Remark 2: Even if $\theta_{n-1} \in \Theta$, the perturbed values $\widetilde{\theta}_n = \theta_{n-1} \pm c_n e_\mu$in the FDSA or $\widetilde{\theta}_n = \theta_{n-1} \pm c_n \Delta_n$in the SPSA case may not lie in Θ. A similar problem may arise for the updated value θ_n. A standard approach to prevent such divergence is to reproject the parameter value inside $\Theta = \prod_{\mu=1}^{m}\left[\theta_\mu^{\min}, \theta_\mu^{\max}\right]$. For the perturbed values, reprojection can be applied by modifying the step size $c_n$accordingly, in both sides of the perturbation. For the parameter update, standard reprojection can be performed.

V. Applications

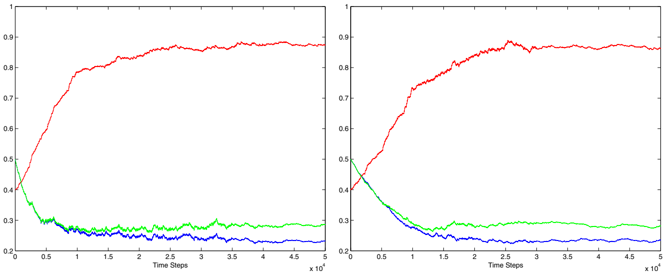
A. Example 1: Linear Gaussian State-Space Model

Let us consider the following linear state space model

$$X_{n+1} = \phi X_n + \sigma_v V_{n+1}, \; X_0 \sim \mathcal{N}\left(0, \frac{\sigma_v^2}{1-\phi^2}\right)$$

$$Y_n = X_n + \sigma_w W_n$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We are interested in estimating the parameter vector $\theta \triangleq \left[\sigma_v, \phi, \sigma_w\right]$. Using $N = 1000$ particles, the true parameter vector was set to $\theta^* \triangleq [0.2, 0.9, 0.3]$ and was initialized at $\theta_0 \triangleq [0.5, 0.4, 0.5]$. Results using the FDSA and the SPSA algorithms are shown in Figure 1. In both examples, our estimates converge to a value $\widehat{\theta}$ in the neighborhood of θ^*.



(a) FDSA estimates (b) SPSA estimates

Fig. 1. On-line parameter estimates $\theta_n = \left[\sigma_{v,n}, \phi_n, \sigma_{w,n}\right]$ for the linear Gaussian model using FDSA and SPSA, with $N = 1000$ particles. From top to bottom: ϕ_n, $\sigma_{w,n}$ and $\sigma_{v,n}$. The true parameters were $\theta^* = [0.2, 0.9, 0.3]$.

B. Example 2 : Stochastic Volatility Model

We consider a discrete-time approximation of a popular diffusion model used in option pricing. The model is given by

$$X_n = \phi X_{n-1} + \sigma V_n, \; X_0 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right)$$

$$Y_n = \beta \exp\left(\frac{X_n}{2}\right) W_n$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and the unknown parameter vector is set to $\theta \triangleq [\sigma, \phi, \beta]$. The true parameter values were chosen to be $\theta^* \triangleq [0.6, 0.9, 0.7]$. Parameter estimation using the SPSA algorithm was performed using $N = 1000$ particles. Figure 2 shows the results. As it can be seen, the algorithm converges towards a value around θ^*. The step sizes were set to $c_n = c_0/n^{0.101}$, with $c_0 = [0.02, 0.01, 0.02]$ and $\gamma = [5, 5, 5] \times 10^{-3}$, where γ was halved every several thousands of iterations.

C. Example 3: A Bimodal Non-linear Model

We use the following standard dynamic model [3], [6]

$$X_n = \theta_1 X_{n-1} + \theta_2 \frac{X_{n-1}}{1 + X_{n-1}^2} + \theta_3 \cos(1.2n) + \sigma_v V_n,$$

$$Y_n = c X_n^2 + \sigma_w W_n,$$

where $\sigma_v^2 = 10$, $c = 0.05$, $\sigma_w = 1$, $X_0 \sim \mathcal{N}(0,2)$, $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. Here we seek

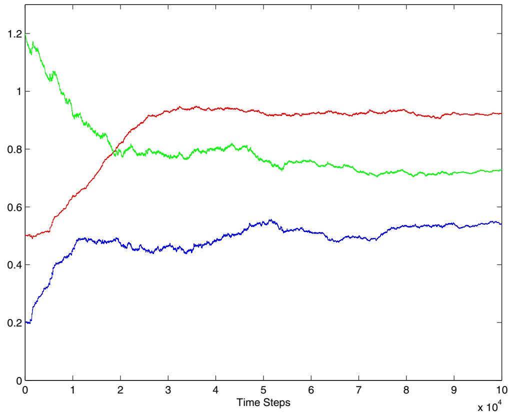


Fig. 2. On-line parameter estimates $\theta_n = [\sigma_n, \phi_n, \beta_n]$ for the Stochastic Volatility model using SPSA and $N = 1000$ particles. From top to bottom: ϕ_n, β_n and σ_n. The true parameters were $\theta^* = [0.6, 0.9, 0.7]$.

maximum likelihood estimates of $\theta = [\theta_1, \theta_2, \theta_3]$. This is a complex problem with a highly multimodal likelihood. Therefore it is important to initialize the algorithm properly, else some of the parameter estimates might get trapped in local maxima. We set the true parameter values to $\theta^* = [0.5, 25, 8,]$ and we initialize at $\theta_0 = [0.2, 20, 5]$. Figure 3 shows the results obtained using SPSA; convergence towards θ^* is again evident. As it was mentioned earlier, the choice of the step sizes is critical to the performance of the SPSA algorithm. In this example this is particularly true due to the difference in the relative sensitivity of the three unknown parameters. The results presented are based on a perturbation step size $c_n = c_0/n^{0.101}$, where $c_0 = [0.02, 2.5, 1] \times 10^{-4}$. For the parameter estimate we have chosen a constant step size $\gamma = [0.004, 6, 15] \times 10^{-4}$.

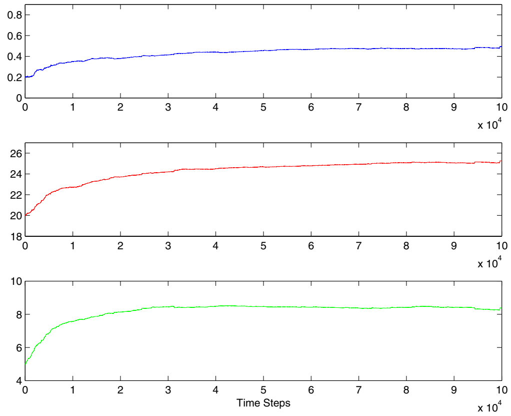


Fig. 3. On-line parameter estimates $\theta_n = [\theta_{1,n}, \theta_{2,n}, \theta_{3,n}]$ for the bimodal nonlinear model using SPSA and $N = 1000$ particles. From top to bottom: $\theta_{1,n}$, $\theta_{2,n}$ and $\theta_{3,n}$. The true parameters were $\theta^* = [0.5, 25, 8]$.

VI. Discussion

In this paper we have proposed fast and simple gradient-free methods to perform on-line maximum likelihood parameter estimation in general state space models, using SMC filters. The methods are based on measurements of the objective function and do not involve any gradient calculations. The SPSA method is particularly attractive over the FDSA due to its reduced computational complexity that remains fixed with the dimensions of the parameter vector. However, its performance is very sensitive to the step size parameters and special care should be taken when these are selected.
Simulation results demonstrate that the methods are effective and at the same time very computationally efficient compared to gradient-based methods. Nevertheless, if one decides to allow for more resources and use a gradient-based approach, the gradient-free algorithms proposed here can still prove extremely useful in exploring the parameter space and choosing suitable initial values for the parameter vector.

References

[1] B.L. Chan, A. Doucet and V.B. Tadić, "Optimisation of particle filters using simultaneous perturbation stochastic approximation", *Proc. IEEE ICASSP,* 2003, pp. 681-684.

[2] D. Bertsekas, *Nonlinear Programming*, 2nd Edition, Athena Scientific, 1999.

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

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

[5] P. Fearnhead, "MCMC, sufficient statistics and particle filter," *J. Comp. Graph. Stat.*, vol. 11, 2002, pp. 848-862.

[6] N.J. Gordon, D.J. Salmond and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, 1993, pp.107-113.

[7] J. Kiefer and J. Wolfowitz, "Stochastic estimation of a regression function," *Ann. of Math. Stat.*, vol. 33, 1952, pp. 462-466.

[8] N.L. Kleinman, J.C. Spall and D.Q. Naiman, "Simulation-based optimisation with stochastic approximation using common random numbers," *Management Science*, vol. 45, no. 11, 1999, pp. 1571-1578.

[9] H.J. Kushner and D.S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Varlag, NY, 1978.

[10] H.J. Kushner and G.G. Yin, *Stochastic Approximation Algorithms and Applications*, Springer-Varlag, NY, 1997.

[11] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Am. Statist. Ass.*, vol. 93, 1998, pp. 1032-1044.

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

[13] L. Ljung and T. Söderström *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, 1983.

[14] G.C. Pflug, *Optimization of Stochastic Models*. Kluwer, 1996.

[15] G. Poyiadjis, A. Doucet and S.S. Singh, "Particle methods for optimal filter derivative: Application to parameter estimation," *Proceedings IEEE ICASSP,* 2005.

[16] G. Poyiadjis, S.S. Singh and A. Doucet, "Novel Particle Filter Methods for recursive and batch parameter estimation in general state space models," Technical Report, CUED/F-INFENG/TR-536, Engineering Department, Cambridge University, 2005.

[17] J.C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans Autom. Control*, vol. 37, 1992, pp. 332-341.

[18] J.C. Spall, "An overview of simultaneous perturbation method for efficient optimisation," John Hopkins APL Technical Digest, vol. 19, no. 4, 1998, pp. 482-492.

[19] J.C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimisation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 4, 1998, pp. 817-823.

[20] J. C. Spall, "Adaptive stochastic approximation by the simultaneous perturbation method," *IEEE Trans. Autom. Contr.,* vol. 45, 2000, pp. 1839–1853.

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

[22] V.B. Tadić and A. Doucet, "Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models," *Stochastic Processes and Their Applications*, vol. 115, 2005, pp. 1408-1436.