ON THE USE OF PARTICLE FILTERING FOR MAXIMUM LIKELIHOOD PARAMETER ESTIMATION

Olivier Cappé and Eric Moulines

Centre National de la Recherche Scientifique & Ecole Nationale Supérieure des Télécommunications
46 rue Barrault, 75634 Paris, France
web: `http://www.tsi.enst.fr/~cappe/`

ABSTRACT

Particle filtering – perhaps more properly named Sequential Monte Carlo – approaches have a strong potential for signal and image processing applications. A problem of great practical significance in this field, which remains largely unsolved as of today, is the estimation of fixed model parameters based on the output of sequential simulations.

In this contribution, we investigate maximum likelihood estimation approaches based either on gradient or EM (Expectation-Maximization) techniques and show that several recently proposed methods share the common feature of requiring the approximation of the expectation of a sum functional of the hidden states, conditionally on all the available observations. Considering this general task, we discuss empirical results concerning the influence of the number of particles and sample size. We also propose a robustification of the basic particle estimator which is based on forgetting ideas.

1. INTRODUCTION

In recent years, sequential Monte Carlo methods have been put in use, sometimes very successfully, in applications as diverse as target tracking, mobile localization, positioning, computer vision and digital communications [8, 13]. It is well-known however that calibration of model parameters based on the output of particle filtering is a difficult issue. Note that although we focus here on likelihood-based methods the same observation holds true in the Bayesian framework.

Although the likelihood of the observations may readily be approximated from the simulated system of particles [12, 11], the use of the obtained approximation for parameter estimation is not trivial. In particular, being able to approximate the value of a function is arguably not sufficient for finding efficiently its maximum, particularly in large dimensional models. In addition, the obtained likelihood approximation is non-smooth due to the Monte Carlo error that affects the likelihood approximations computed for each value of the parameters. Note that it is still the case, even when cleverly fixing the random seeds used for simulating the system of particles as proposed by [12], due to the fundamentally non-smooth nature of the resampling operator.

We consider below the approximation of quantities which are instrumental in effective maximization of the likelihood either when using EM-based or gradient-based methods. Our main concern will be to approximate reliably such quantities using particle filtering methods. Note that we don't consider specifically the task of recursive or on-line estimation although the approaches discussed here could obviously be adapted for this purpose.

2. THE MODEL

We focus on particle filtering techniques in the context of hidden Markov models although the techniques discussed here also apply for more general models such as Markov switching autoregressive models [2]. A hidden Markov model is such that

1. $\{X_k\}_{k\geq 0}$ is Markovian with initial distribution q_0 and transition density function q such that, for any function f, $\mathrm{E}[f(X_0)] = \int f(x)\, q_0(x)dx$ and
$$\mathrm{E}[f(X_n)|X_{0:n-1}] = \int f(x)\, q(X_{n-1},x)dx$$
where $X_{0:n-1}$ denotes the collection of variables $X_0,\ldots,X_{n-1}$.
2. $\{Y_k\}_{k\geq 0}$ is conditionally independent given $\{X_k\}_{k\geq 0}$ with (marginal) transition density function g such that, for arbitrary functions $f_0,\ldots,f_n$,
$$\mathrm{E}\left[\prod_{k=0}^{n} f_k(Y_k)\,\middle|\, X_{0:n}\right] = \prod_{k=0}^{n}\int f_k(y)\, g(X_k,y)dy$$
where g is sometimes referred to as the *(conditional) likelihood function.* In the following, we always consider g as a function of its first argument only and write $g_k(x) = g(x,Y_k)$ for the conditional likelihood function evaluated in Y_k.

Using Bayes' rule it is easily verified that – see e.g.[9] – the *joint smoothing density*, which is the main quantity of interest in particle filtering, is given by

$$\phi_{0:n|n}(x_{0:n}) = \mathrm{L}_n^{-1} q_0(x_0) g_0(x_0) \prod_{k=1}^{n} q(x_{k-1},x_k) g_k(x_k) \qquad (1)$$

where the normalization factor L_n is the *likelihood* of the observations $Y_{0:n}$. From (1) we may derive the recursive formulation

$$\phi_{0:n|n}(x_{0:n}) = c_n^{-1}\phi_{0:n-1|n-1}(x_{0:n-1}) q(x_{n-1},x_n) g_n(x_n) \qquad (2)$$

where $c_n = \mathrm{L}_n/\mathrm{L}_{n-1}$. Marginalizing (2), taking the log and summing the obtained expression for all indices between 0 and n yields the well-known expression of the likelihood

$$\ell_n \stackrel{\mathrm{def}}{=} \log \mathrm{L}_n = \sum_{k=0}^{n} \log c_k = \log \int q_0(x) g_0(x) + \sum_{k=1}^{n} \log \iint \phi_{k-1}(x) q(x,x') g_k(x') dx dx' \qquad (3)$$

where ϕ_k is the *filtering* density (marginal of $\phi_{0:k|k}$ in x_k).

Particle filtering, in its most basic form (also known as *sequential importance sampling with resampling*), consists in approximating these exact smoothing relations by propagating particle trajectories in the state space of the hidden chain according to

- At time 0, draw N particles $\{\xi_0^i\}_{1\leq i\leq N}$ from a common probability density ρ_0 and compute the importance weight $\omega_0^i = q_0(\xi_0^i) g_0(\xi_0^i)/\rho_0(\xi_0^i)$.
- For successive time indices, simulate ξ_{k+1}^i from a transition density function $r(\xi_k^i,\cdot)$ and update the weight according to $\omega_{k+1}^i = \omega_k^i q(\xi_k^i,\xi_{k+1}^i) g_{k+1}(\xi_{k+1}^i)/r(\xi_k^i,\xi_{k+1}^i)$ and the trajectory by $\xi_{0:k+1}^i = (\xi_{0:k}^i,\xi_{k+1}^i)$.

- From time to time resample by first normalizing the weights $\{\omega_k^i\}_{1\le i\le N}$ by their sum so that $\sum_{i=1}^N \omega_k^i = 1$ and then drawing indices $I_k^1,\dots,I_k^N$ in $\{1,\dots,N\}$ such that

$$\mathrm{E}(\#\{1\le j\le N : I_k^j = i_0\}) = N\omega_k^{i_0}$$

the weights are all reset to a common value and the trajectory is updated according to

$$\xi_{0:k}^j = \xi_{0:k}^{I_k^j}$$

Note that due to resampling, the notation ξ_k^i becomes somewhat ambiguous and we will use the notation $\xi_{0:k}^i(l)$ to denote the point of index l in the ith particle trajectory at index k (with $k\ge l$). By convention, $\xi_{0:k}^i(k)$ is denoted simply by ξ_k^i.

At any index k, the self-normalized estimate

$$\sum_{i=1}^N \frac{\omega_k^i}{\sum_{j=1}^N \omega_k^j} f(\xi_{0:k}^i) \tag{4}$$

is an approximation of $\mathrm{E}[\,f(X_{0:k})\,|\,Y_{0:k}]$ (where f is an arbitrary integrable function of the $k+1$ first state variables). Compared to other more ad-hoc alternatives, particle filtering can be shown to be convergent (in a suitable probabilistic sense) as the number N of particles increases [8, 5, 6, 2].

3. APPROACHES FOR MAXIMUM LIKELIHOOD ESTIMATION

We now assume that the model characteristics q, g and q_0 depend on a parameter vector θ. We will use θ as a superscript to indicate which quantities depend on the parameter. Since we consider iterative maximization algorithms there are always two values of the parameters which play a role at any given iteration: the current one and its update. To avoid notational blow up *we omit the iteration index and make dependence with respect to the current value of the parameter implicit*, for instance, q refers to current estimate of the hidden chain's transition density, while q^θ refers to the same quantity for a different value of the parameter.

For doing maximum likelihood estimation, one needs to estimate quantities of the form

$$\tau_n = \mathrm{E}\left[\sum_{k=0}^{n-1} s_k(X_k, X_{k+1}) \,\middle|\, Y_{0:n}\right] \tag{5}$$

where s_0 to s_n are, possibly vector-valued, functions. This is obvious in the context of the EM algorithm [7] since the joint density of the states and observations $\log p^\theta(x_{0:k}, y_{0:k})$ has precisely the form given in (5) with $s_k(x_k,x_{k+1}) = \log q^\theta(x_k,x_{k+1}) + \log g_k^\theta(x_k+1)$ for $k\ge 1$ and $s_0(x_0) = \log g_0^\theta(x_0)$ (assuming that q_0 does not depend on the parameter θ).

It is also true with gradient based method since Fisher's identity [7] states that the gradient of the log-likelihood ℓ_n defined in (3), evaluated at the current value of the parameter, is obtained when $s_k(x_k,x_{k+1}) = \nabla\log q(x_k,x_{k+1}) + \nabla\log g_k(x_k+1)$ (for $k\ge 1$) and $s_0(x_0) = \nabla\log g_0(x_0)$, where ∇ denotes the gradient taken at the current value of the parameter. The approach followed in [4, 10] to derive the gradient approximation is slightly different since it is based on the so-called *sensitivity equations* obtained by differentiating the filtering recursion with respect to the parameter. It can however be shown that the obtained recursion corresponds to a recursive rewriting of Fisher's formula and hence is equivalent to the formula given here [2]. Note also that (5) is obviously very different from the expression of the log-likelihood in (3): equation (5) does implies the joint smoothing distribution while (3) may be written as a telescoping sum of terms, each of which only involves the filtering distribution.

Quite naturally the approximation of (5) based on (4) then consists in propagating a system of particle trajectories and associated weights under the current value of the parameter, which we denote by $\{\xi_{0:n}^i, \omega_n^i\}_{1\le i\le N}$, and using the estimator

$$\hat\tau_n^N = \sum_{i=1}^N \frac{\omega_n^i}{\sum_{j=1}^N \omega_n^j} \sum_{k=0}^{n-1} s_k\left(\xi_{0:n}^i(k), \xi_{0:n}^i(k+1)\right) \tag{6}$$

It is straightforward to verify that storing the whole particle trajectories is indeed not required to evaluate (6): upon defining $\gamma_k^i = \sum_{l=0}^{k-1} s_l\left(\xi_{0:k}^i(l), \xi_{0:k}^i(l+1)\right)$ (for $k\ge 1$), we have

$$\gamma_{k+1}^i = \begin{cases} \gamma_k^i + s_k(\xi_k^i, \xi_{k+1}^i) & \text{in the abscence of resampling} \\ \gamma_k^{I_{k+1}^i} + s_k(\xi_k^{I_{k+1}^i}, \xi_{k+1}^{I_{k+1}^i}) & \text{when resampling occurs} \end{cases} \tag{7}$$

where $\hat\tau_n^N$ is obtained as $\sum_{i=1}^N \omega_n^i\gamma_n^i$. Hence we only need to store for each particle, its current position ξ_k^i, weight ω_k^i and partial cumulated sum γ_k^i. The method thus necessitates only minor adaptations once the particle filter has already been implemented. Note that this algorithm may also be obtained directly as a particle approximation of an equivalent recursive (in n) formulation of (5) [1].

4. APPROXIMATION OF SMOOTHED SUM FUNCTIONALS

The main question to be answered is to determine how good is the approximation of (5) given in (6) for different values of N and n? Although we cannot actually provide a full theoretical analysis of this issue, we give a number of hints in that direction, based on our experience of this approach in several examples, one of which will be discussed below.

1. Although the variance of $\hat\tau_n^N$ grows as n increases (while keeping the number N of particles fixed) it does not blow up, as sometimes conjectured; note that normalization issues are important here since τ_n obviously is a quantity that increases with n.
2. As expected, the approximation improves with larger values of N (the number of particles) but, when N is sufficiently large, it is possible to modify slightly $\hat\tau_n^N$ so as to reduce significantly its variance (at the price of introducing a, usually, negligible bias).

The first point in particular is somewhat surprising because τ_n as defined in (5) depends on the complete trajectory between indices 0 and n and it seems that the approximation $\hat\tau_n^N$, with a fixed value of N, could get arbitrarily bad as n increases. Typically, this is not the case and the rate of increase of the approximation error with n is very moderate. This is particularly true when τ_n is properly normalized: τ_n is a persistent sum (i.e. consisting of a number of terms which increase as n grows) whose value is always used normalized by the number of processed observations (see example in Section 5 below). Under this normalization, the increase of the approximation error with n is not very significant (see Figure 2).

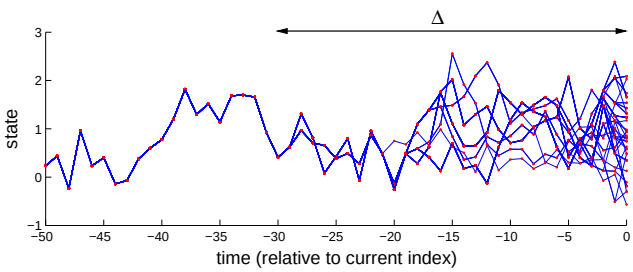

Figure 1: Typical particles trajectories for $N = 50$ (see Section 5 for details of model and algorithm).

We now discuss some elements that justify the construction which makes it possible to reduce the variability of the particle approximation. To simplify the discussion, we consider in the sequel

that resampling is used systematically (at each time index) so that $\omega_k^i = 1/N$ for all k and i: in particular, the weights in (6) are all equal to $1/N$ and γ_k^i is systematically updated according to the second option in (7).

It is important to understand that as n increase, there is indeed a part of the sum in (6) that doesn't get updated anymore due to the successive resamplings: there exists (with high probability) a finite random delay Δ_k (which may vary with k) defined as the smalled integer for which $\xi_{0:k}^i(k-\Delta_k+1) = \xi_{0:k}^j(k-\Delta_k+1)$ for all $i, j \in \{1,\ldots,N\}$, i.e. all active trajectories at index k have a common ancestor back in the past at index $k-\Delta_k+1$ (see Figure 1 for an illustration). It is not hard to see that for any $n \geq k$ and $l \leq k-\Delta_k+1$, $\xi_{0:n}^i(l) = \xi_{0:k}^j(l)$. Thus the contribution of index l (for $l \leq k-\Delta_k$) is fixed to $s_l\left(\xi_{0:k}^i(l), \xi_{0:k}^i(l+1)\right)$ for all later indices $n \geq k$ (as well as for any i, since all trajectories have collapsed into a single one at this stage).

Because the collapsing times Δ_k are random it is very hard to tell anything about the distribution of $\xi_{0:k}^j(k-\Delta_k+1)$. On the other hand, if Δ_k was deterministic and equal to δ, then $s_{k-\delta}\left(\xi_{0:k}^i(k-\delta), \xi_{0:k}^i(k-\delta+1)\right)$ would be and estimator of

$$\mathrm{E}\left[s_{k-\delta}(X_{k-\delta}, X_{k-\delta+1}) \,|\, Y_{0:k}\right]$$

that is the expectation of $s_{k-\delta}$ under the δ-lag smoothing distribution. This remark naturally suggests the following question: is the δ-lag smoothing distribution $\mathrm{P}(X_{k-\delta} \in \cdot, X_{k-\delta+1} \in \cdot \,|\, Y_{0:k})$ close to $\mathrm{P}(X_{k-\delta} \in \cdot, X_{k-\delta+1} \in \cdot \,|\, Y_{0:n})$ for $n \geq k$? This question is connected to a more general concern known as *forgetting properties* of the smoothing and filtering distributions. Very schematically the answer to this question is known (empirically) to be yes in many situations but is fairly hard to establishes on solid theoretical grounds. In all models for which the stability (in n) of the particle filter has currently been proved however, the answer can be shown to be "yes, at a rate which is exponential in the lag δ" [5, 6, 2].

These arguments suggest that waiting for all the trajectories to collapse – as (7) implies – is not a very efficient simulation principle. Hence when N is sufficient so that forgetting occurs for values of δ which may be far smaller than typical values of Δ_n it is more appropriate to impose the lag δ after which resampling of the past trajectories is inhibited. To do this, we modify the definition of τ_n in (5) into

$$\tau_n^\delta = \sum_{k=0}^{n-1} \mathrm{E}\left[s_k(X_k, X_{k+1}) \,|\, Y_{0:(k+\delta)\wedge n}\right] \tag{8}$$

which may be approximated by

$$\hat{\tau}_n^{\delta,N} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{n} s_k\left(\xi_{0:(k+\delta)\wedge n}^i(k), \xi_{0:(k+\delta)\wedge n}^i(k+1)\right) \tag{9}$$

Although a little bit more involved than in the case of (6), (9) may be updated recursively by maintaining a cache of the recent history of the particles $\{\xi_{0:n}^i(n-\delta+1:n)\}_{1\leq i\leq n}$ as well as the cumulated contribution of terms that will not get updated anymore

$$\sum_{i=1}^{N}\left\{s_{n-\delta}\left(\xi_{0:n-1}^{I_n^i}(n-\delta), \xi_{0:n-1}^{I_n^i}(n+1-\delta)\right) + \sum_{k=0}^{n-\delta-1} s_k\left(\xi_{0:k+\delta}^i(k), \xi_{0:k+\delta}^i(k+1)\right)\right\}$$

Apart from increased storage requirements, computing the reduced-lag approximation $\hat{\tau}_n^{\delta,N}$ is clearly not computationally more demanding than computing τ_n^δ.

5. SOME RESULTS

For illustration purposes we consider a one dimensional model for which exact computation is also available, namely a noisily observed Gaussian AR(1) model such that $q^\theta(x,x') = \mathrm{N}(x'; \phi x, \sigma^2)$, $g^\theta(x,y) = \mathrm{N}(y; x, \rho^2)$, where $\mathrm{N}(\cdot; \mu, \upsilon)$ denotes the Gaussian pdf (probability density function) with mean μ and variance υ. The initial pdf q_0 is an improper constant (also called diffuse) prior; meaning in particular that the initial filtering pdf $\phi_0(x)$ is given by $\mathrm{N}(x; Y_0, \rho^2)$. Throughout this section we use a simulated dataset of length $n = 500$ with parameters $\phi = 0.98$, $\sigma = 0.2$, $\rho = 1$.

It is straightforward to check that the EM algorithm applied to this model requires the approximation of

$$\tau_{n,1} = \mathrm{E}\left[\sum_{k=0}^{n-1} X_k^2 \,\middle|\, Y_{0:n}\right]$$

$$\tau_{n,2} = \mathrm{E}\left[\sum_{k=0}^{n-1} X_k X_{k+1} \,\middle|\, Y_{0:n}\right]$$

$$\tau_{n,3} = \mathrm{E}\left[\sum_{k=1}^{n} X_k^2 \,\middle|\, Y_{0:n}\right]$$

$$\tau_{n,4} = \mathrm{E}\left[\sum_{k=0}^{n} (Y_k - X_k)^2 \,\middle|\, Y_{0:n}\right]$$

and updates the parameters according to

$$\phi = \tau_{n,2}/\tau_{n,1} \tag{10}$$

$$\sigma^2 = (\tau_{n,3} - \phi\,\tau_{n,2})/n \tag{11}$$

$$\rho^2 = \tau_{n,4}/(n+1) \tag{12}$$

As mentioned before, the above update equations depend on the expected sums $\tau_{n,1}, \ldots, \tau_{n,4}$ normalized by terms of order n (or by another $\tau_{n,j}$ which is equivalent) rather than on the expected sums themselves.

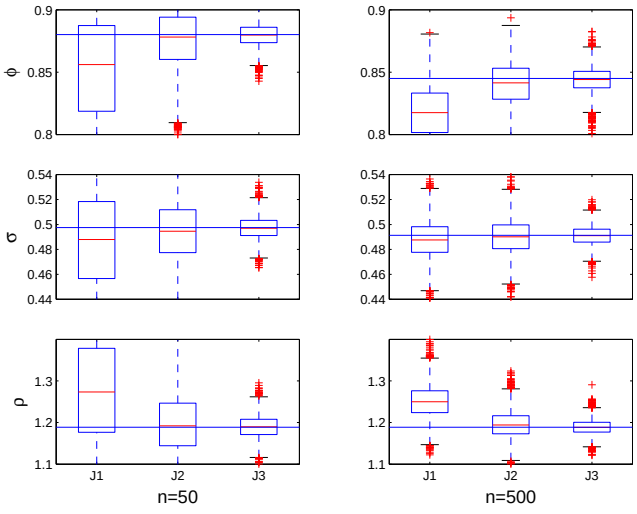

Figure 2: Box-and-whiskers plots of the approximate parameter updates for two different observations lengths compared with exact update (solid horizontal line); **J1:** $N = 10$, **J2:** 100, **J3:** 1000 particles (plot based on 5000 independent replications of the particles).

Figure 2 displays the parameter updates (10)–(12) based on sequential Monte Carlo approximations of $\tau_{n,1}$ to $\tau_{n,4}$ (here we used the simple bootstrap filter with systematic resampling) using the basic approach described in (7) and compared to the exact EM updates. The values $\phi = 0.8$, $\sigma = 0.5$ and $\rho = 2$ are used as current

estimates of the parameter (which are thus rather far from the actual maximum likelihood estimate, both when $n = 50$ and $n = 500$). Note that when the number of particles is really too small ($N = 10$) we observe a bias for some parameters which is coherent with the previous discussion (knowing that when $N = 10$ the average values of Δ_n is indeed not much larger than 10).

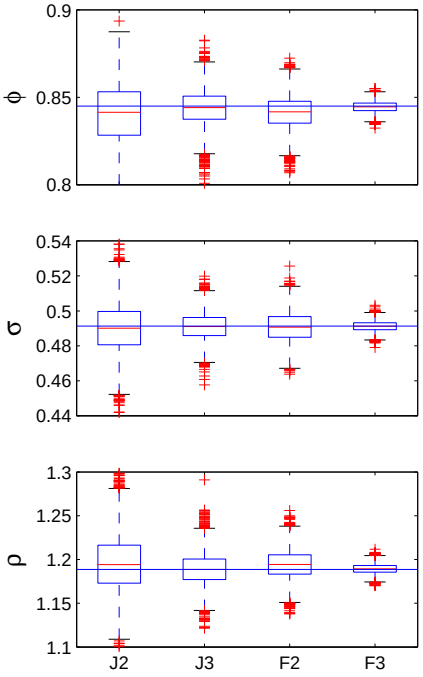

Figure 3: Same display as in Figure 2 for $n = 500$; comparison of "joint" approximation $\hat{\tau}_n^N$ with **J2:** $N = 100$ and **J3:** 1000 particles with "fixed-lag" approximation $\hat{\tau}_n^{\delta,N}$ with **F2:** $N = 100$ and **F3:** 1000 particles and lag $\delta = 20$.

In Figure 3, we report only the case $n = 500$ and compare the basic approximation strategy with the one based on fixed-lag smoothing with $\delta = 20$, which is a reasonable value of the lag for which forgetting is already quite strong in this example. It is obvious that fixed-lag smoothing drastically reduce the variance without significantly raising the bias: boxes labelled "F2" (fixed-lag smoothing) are roughly equivalent to those labelled "J3" (joint smoothing) which are obtained with 10 times more particles.

In any case, the approximations obtained using sequential Monte Carlo appear sufficiently reliable (even for large values of n) to be used in simulation-based maximum-likelihood algorithms such as Monte Carlo EM. To give an idea of the type of results that one may obtain we show in Figure 4 the first fifty iterations of the SEM (Stochastic EM) algorithm of [3] compared to the exact EM trajectory. Here the SEM algorithm simply consists in using (10)–(12) with the particle approximations substituted for the exact expectations. Although SEM is not an algorithm that converges to the maximum of the likelihood in general, it is clear that it does reasonably well in this case, even with a very moderate number of particles ($N = 25$) – see also [2] for other examples.

We currently believe that a complete theoretical study of the behavior of estimates based on the fixed-lag approximation is possible using available results on the convergence of the particle filter and stochastic versions of the EM algorithm.

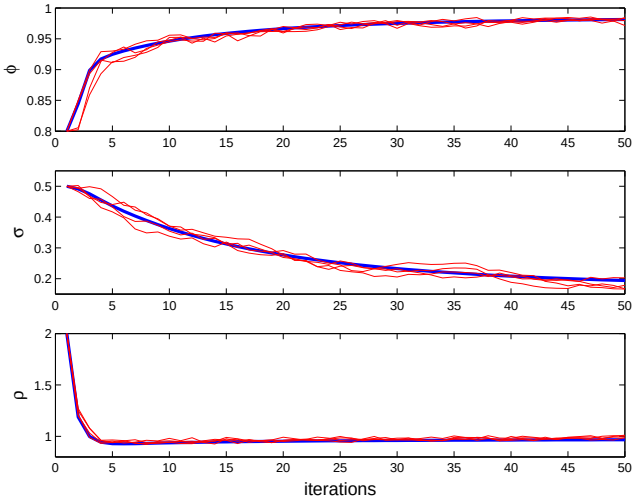

Figure 4: Four independent trajectories of the SEM algorithm with $N = 25$ particles superimposed on the exact EM trajectory (bold line). Same data as in previous figures, with $n = 500$.

REFERENCES

[1] O. Cappé. Recursive computation of smoothed functionals of hidden Markovian processes using a particle approximation. *Monte Carlo Methods Appl.*, 7(1-2):81–92, 2001.

[2] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

[3] G. Celeux and J. Diebolt. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Comput. Statist.*, 2:73–82, 1985.

[4] F. Cérou, F. Le Gland, and N. Newton. Stochastic particle methods for linear tangent filtering equations. In J.-L. Menaldi, E. Rofman, and A. Sulem, editors, *Optimal Control and PDE's - Innovations and Applications, in Honor of Alain Bensoussan's 60th Anniversary*, pages 231–240. IOS Press, 2001.

[5] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.*, 50(3):736–746, 2002.

[6] P. Del Moral. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38 (with discussion), 1977.

[8] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[9] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte-Carlo sampling methods for Bayesian filtering. *Stat. Comput.*, 10:197–208, 2000.

[10] A. Doucet and V. B. Tadić. Parameter estimation in general state-space models using particle methods. *Ann. Inst. Statist. Math.*, 55(2):409–422, 2003.

[11] M. Hürzeler and H. R. Künsch. Monte Carlo approximations for general state-space models. *J. Comput. Graph. Statist.*, 7:175–193, 1998.

[12] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *J. Am. Statist. Assoc.*, 94(446):590–599, 1999.

[13] B. Ristic, M. Arulampalam, and A. Gordon. *Beyond Kalman Filters: Particle Filters for Target Tracking*. Artech House, 2004.