# EFFICIENT PARTICLE-BASED ONLINE SMOOTHING IN GENERAL HIDDEN MARKOV MODELS

Johan Westerborn, Jimmy Olsson\*

Royal Institute of Technology Department of mathematics KTH SE-100 44 Stockholm Sweden

# ABSTRACT

This paper deals with the problem of estimating expectations of sums of additive functionals under the joint smoothing distribution in general hidden Markov models. Computing such expectations is a key ingredient in any kind of expectation-maximization-based parameter inference in models of this sort. The paper presents a computationally efficient algorithm for online estimation of these expectations in a forward manner. The proposed algorithm has a linear computational complexity in the number of particles and does not require old particles and weights to be stored during the computations. The algorithm avoids completely the well-known particle path degeneracy problem of the standard forward smoother. This makes it highly applicable within the framework of online expectation-maximization methods. The simulations show that the proposed algorithm provides the same precision as existing algorithms at a considerably lower computational cost.

*Index Terms*— Hidden Markov models, particle filters, smoothing methods, Monte Carlo methods, state estimation

#### 1. INTRODUCTION

A hidden Markov model (HMM) is a bivariate model consisting of an observable process  $\{Y_t\}_{t=0}^{\infty}$  and an unobservable Markov chain  $\{X_t\}_{t=0}^{\infty}$  taking values in some general state spaces  $\mathcal{Y}$  and  $\mathcal{X}$ , respectively. Conditionally on the unobserved process the observations are assumed to be independent with conditional distribution of  $Y_t$  depending on  $X_t$  only. In this paper we focus on computing smoothed expectations  $\mathbb{E}[S_T(X_{0:T})|Y_{0:T}]$  for additive functionals on the form

$$S_T(x_{0:T}) = \sum_{t=0}^{T-1} s_t(x_t, x_{t+1}), \tag{1}$$

where  $\{s_t\}_{t\geq 0}$  is a sequence of measurable functions on the product space  $\mathcal{X}^2$ . Since the joint distribution of the observed and unobserved process in an HMM usually belongs to the exponential family, computing smoothed expectations of the mentioned form is typically a key ingredient when casting the problem of computing the *maximum likelihood estimator* into the framework of the *expectationmaximization* (EM) algorithm; see e.g. [1, chap. 11]. In this paper we will propose an online algorithm for estimation of such expectations. In particular, the algorithm can with advantage be combined with online implementations of EM; see [2, 3, 4, 5].

# 2. PRELIMINARIES

Throughout this paper we assume that we are given a fixed sequence  $\{y_t\}_{t\geq 0}$  of observations and write, for  $s \leq t$ ,  $y_{s:t} = (y_s, y_{s+1}, \ldots, y_t)$  (this will be our general notation for vectors). Let  $q(x_{t-1}, x_t)$  denote the transition density of the Markov chain,  $g(x_t, y_t)$  the observation transition density and  $\chi$  the initial distribution of the Markov chain. For ease of notation we expunge the dependence on the observations from the notation and let  $g_t(x_t) = g(x_t, y_t)$ . In a parameter estimation framework all these quantities depend on some unknown parameter vector  $\theta$  to be estimated.

We are interested in the joint distribution  $\phi_{0:T|T}$  of a given set  $X_{0:T}$  of latent states given the corresponding observations  $Y_{0:T}$ . This distribution is called the *joint smoothing distribution* and is given by

$$\begin{split} \phi_{0:T|T}(dx_{0:T}) &= L_T^{-1}\chi(x_0)g(x_0, y_0) \\ &\times \prod_{t=1}^T q(x_{t-1}, x_t)g_t(x_t)dx_{0:T} \end{split}$$

where  $L_T$  is the likelihood of the observations defined by

$$L_T = \int \cdots \int \chi(x_0) g(x_0, y_0) \prod_{t=1}^T q(x_{t-1}, x_t) g_t(x_t) dx_{0:T}.$$

For a function  $S_T$  let  $\phi_{0:T|T}(S_T) = \int h(x_{0:T})\phi_{0:T|T}(dx_{0:T})$  be the expectation of  $S_T$  under the distribution  $\phi_{0:T|T}$ .

Throughout this paper we will work with *sequential Monte Carlo* methods, or *particle filters*, to estimate sequentially sequences of distributions. Particle filters can be described as a way of generating sequentially, using importance sampling and resampling techniques, particles and associated importance weights  $\{(\xi_t^i, \omega_t^i)\}_{i=1}^N$  targeting a sequence of distributions. As a key ingredient in our algorithm we will use particle filters to estimate the flow of *filter distributions*  $\phi_t = \phi_{t:t|t}$  using the estimators

$$\phi_t^N(dx_t) = \sum_{i=1}^N \frac{\omega_t^i}{\Omega_t} \delta_{\xi_t^i}(x_t) dx_t,$$
(2)

where  $\Omega_t = \sum_{i=1}^N \omega_t^i$  and  $\delta_{\xi_t^i}(x_t)$  is a unit mass at  $x_t = \xi_t^i$ . There are several algorithms available for producing such sequences of weights and particles where the most simple one is the bootstrap particle filter [6], described in Algorithm 1. These algorithms provide an estimate of  $\phi_{0:T|T}$  as a by-product when looking at the genealogical trajectories  $\xi_{0:T}^i$  of the particles. The resampling operation of the

<sup>\*</sup> Supported by the Swedish Research Council, Grant 2011-5577.

algorithm will collapse these trajectories over time, which makes this method, in its basic form, incapable of estimating the joint smoothing distribution. This problem is known as the degeneracy of the genealogical tree; see [7] [8, sec 14.3.6] for some discussion.

We will now set focus on estimating  $S_T \triangleq \phi_{0:T|T}(S_T)$ , where  $S_T(x_{0:T})$  is of the form (1). The quantity  $S_T$  will be referred to as a *smoothed additive functional*.

Algorithm 1 Bootstrap particle filter 1: for  $i = 1 \rightarrow N$  do simulate  $\xi_0^i \sim \chi$ 2: 3: set  $\omega_0^i \leftarrow g_0(\xi_0^i)$ 4: end for 5: for  $t = 0 \to T - 1$  do for  $i = 0 \rightarrow N$  do 6: Set  $I_t^i = \ell$  w.pr.  $\omega_t^{\ell} / \Omega_t$ 7:  $\begin{array}{l} \text{Simulate } \xi_{t+1}^i \sim q(\xi_t^{I_t^i}, \cdot) \\ \text{Set } \omega_{t+1}^i \leftarrow g_{t+1}(\xi_{t+1}^i) \end{array}$ 8: 9: 10: end for 11: end for

# 3. PREVIOUS WORK

To obtain an estimate of the joint smoothing distribution we work with the *forward-backward decomposition* see e.g. [1, sec 3.3.2]. The forward-backward decomposition is based on the fact that the hidden chain is still Markov when evolving conditionally on the observations. For this purpose, introduce the *backward kernel*  $B_{\phi_t}$  with transition density

$$B_{\phi_t}(x_{t+1}, x_t) \triangleq \frac{\phi_t(x_t)q(x_t, x_{t+1})}{\int \phi_t(x')q(x', x_{t+1})dx'}$$

which can be seen as the distribution of  $X_t$  conditionally on  $X_{t+1}$  as well as the full observation record  $Y_{0:T}$  (hence the name backward kernel). These kernels allows the joint smoothing distribution to be expressed using the backward decomposition

$$\phi_{0:T|T}(dx_{0:T}) = \prod_{u=0}^{T-1} B_{\phi_u}(x_{u+1}, dx_u) \phi_T(x_T).$$

By plugging the particle estimate  $\phi_t^N$  in (2) into the expression of the backward kernel we obtain the particle estimate

$$B_{\phi_t^N}(x_{t+1}, x_t) = \sum_{i=1}^N \frac{\omega_t^i q(\xi_t^i, x)}{\sum_{\ell=1}^N \omega_t^\ell q(\xi_t^\ell, x)} \delta_{\xi_t^i}(x_t)$$

of  $B_{\phi_t}(x_{t+1}, x_t)$ , which lead us to the *forward-filtering backward* smoothing (FFBSm) estimator [9, 10, 11]

$$\phi_{0:T|T}^{N}(dx_{0:T}) = \sum_{i_{0}=1}^{N} \cdots \sum_{i_{T}=1}^{N} \left( \prod_{u=0}^{T-1} \frac{\omega_{u}^{i_{u}}q(\xi_{u}^{i_{u}}, \xi_{u+1}^{i_{u+1}})}{\sum_{\ell=1}^{N} \omega_{u}^{\ell}q(\xi_{u}^{\ell}, \xi_{u+1}^{i_{u+1}})} \right) \\ \times \frac{\omega_{T}^{i_{T}}}{\Omega_{T}} \delta_{(\xi_{0}^{i_{0}}, \dots, \xi_{T}^{i_{T}})}(x_{0:T}) dx_{0:T}.$$
(3)

When working with additive functionals it is possible to achieve a forward-only implementation of the FFBSm [2] by introducing the auxiliary function  $\mathcal{T}_t(x_t) \triangleq \int \cdots \int \prod_{u=0}^{t-1} B_{\phi_u}(x_{u+1}, dx_u) S_t(x_{0:t})$  such that  $\phi_{0:t|t}(S_t) = \int \mathcal{T}_t(x_t)\phi_t(dx_t)$ . The idea is now to update  $\mathcal{T}_t$  using the forward recursion

$$\mathcal{T}_t(x_t) = \int (\mathcal{T}_{t-1}(x_{t-1}) + s_{t-1}(x_{t-1}, x_t)) B_{\phi_{t-1}}(x_t, dx_{t-1})$$

By plugging the particle approximation of the backward kernel into this recursion we may approximate  $T_t(x_t)$  by

$$\mathcal{T}_{t}^{N}(x_{t}) = \sum_{i=1}^{N} \frac{\omega_{t-1}^{i}q(\xi_{t-1}^{i}, x_{t})}{\sum_{\ell=1}^{N} \omega_{t-1}^{\ell}q(\xi_{t-1}^{\ell}, x_{t})} \times \left(\mathcal{T}_{t-1}^{N}(\xi_{t-1}^{i}) + s_{t-1}(\xi_{t-1}^{i}, x_{t})\right),$$

yielding the estimate

$$\mathcal{S}_T^N = \Omega_T^{-1} \sum_{i=1}^N \omega_T^i \mathcal{T}_T^N(\xi_T^i)$$

of  $\mathcal{S}_T$ .

These two algorithms require the normalizing constant

 $\sum_{i_t=1}^{N} \omega_t^{i_t} q(\xi_t^{i_t}, \xi_{t+1}^{i_{t+1}})$  to be computed for each index  $i_{t+1} = 1, \ldots, N$  by summing over all indices  $i_t$ . As a consequence, each iteration of these algorithms has an  $\mathcal{O}(N^2)$  computational complexity.

Introduce the  $\sigma$ -field  $\mathcal{F}_t^N = \sigma\{Y_{0:T}, (\xi_t^i, \omega_t^i); 0 \le t \le T, 1 \le i \le N\}$ . Then the FFBSm estimator can, conditioned on  $\mathcal{F}_t^N$ , be viewed as a probability distribution on the indices. Consider for  $t \in \{0, \ldots, T-1\}$  the Markov transition matrix  $\{\Lambda_t^N(i, j)\}_{i,j=1}^N$  with elements given by

$$\Lambda_t^N(i,j) = \frac{\omega_t^j q(\xi_t^j, \xi_{t+1}^i)}{\sum_{\ell=1}^N \omega_t^\ell q(\xi_t^\ell, \xi_{t+1}^i)}$$

A set of particles trajectory being approximately distributed according to the smoothing distribution can now be produced by first drawing  $J_T$  such that  $\mathbb{P}(J_T = j_T | \mathcal{F}_T^N) \propto \omega_T^{j_T}$  and then, recursively for  $t = T - 1, \ldots, 0$ , drawing backwards  $J_t$  conditioned on  $J_{t+1}$ according to  $\mathbb{P}(J_t = j_t | J_{t+1} = j_{t+1}, \mathcal{F}_T^N) = \Lambda(j_{t+1}, j_t)$ . After this, return  $(\xi_0^{J_0}, \xi_1^{J_1}, \ldots, \xi_T^{J_T})$  as a sample from the joint smoothing distribution. This algorithm is called the *forward-filtering backward simulation* (FFBSi) algorithm [12]. As in the FFBSm algorithm we require, when drawing the indices  $J_t$ , the normalizing constant  $\sum_{\ell=1}^N \omega_t^\ell q(\xi_t^\ell, \xi_{t+1}^{i_t+1})$  to be computed. Thus, we are still left with an algorithm having an  $\mathcal{O}(N^2)$  complexity.

However, as found in [11], a faster version of the FFBSi algorithm can be obtained under the rather mild assumption that there exists a constant  $q_+$  such that  $q(x, y) \leq q_+$  for all  $(x, y) \in \mathcal{X}^2$ . In this case it is possible to apply an accept-reject sampling scheme by drawing a candidate  $J_t^{(k)}$  with a probability proportional to  $\omega_t^i$  and then accepting this candidate with the probability  $q(\xi_t^{J_t^{(k)}}, \xi_{t+1}^{J_{t+1}})/q_+$ . If rejected, a new candidate is drawn. The first index  $J_T$  is drawn as before. The expected amount of draws needed for the accept-reject scheme will, as N tends to infinity, tend to a constant that does not depend on N. Consequently the computational complexity of the algorithm will be of  $\mathcal{O}(N)$  [11]. However, the FFBSi algorithm can only be implemented in batch mode by first running the particle filter up to the final time point T and, after this, sampling the backward chain backwards to the initial time point. Nevertheless, in the next section we present a novel algorithm that can be viewed as a forward-only—online—implementation of the rapid version of FFBSi.

#### 4. NEW ALGORITHM

Our new algorithm relies on the same decomposition as the forwardonly version of the FFBSm algorithm described above. Given that we have estimates  $\tilde{\mathcal{T}}_{t-1}^{N}(\xi_{t-1}^{i})$  of  $\mathcal{T}_{t-1}(\xi_{t-1}^{i})$  for  $i = 1, \ldots, N$  and particles and weights  $\{(\xi_{t-1}^i, \omega_{t-1}^i)\}_{i=1}^N$  estimating the filtering distribution  $\phi_{t-1}$ , the algorithm first propagates the particle filter one step resulting in new particles and weights  $\{(\xi_t^i, \omega_t^i)\}_{i=1}^N$  estimating the filter distribution  $\phi_t$ . For each particle  $\xi_t^i$  we draw K > 1 indices  $J_t^{i,k}$  according to

$$\mathbb{P}\left(J_{t}^{i,k} = j|\mathcal{F}_{t}^{N}\right) = \frac{\omega_{t-1}^{j}q(\xi_{t-1}^{j},\xi_{t}^{i})}{\sum_{\ell=1}^{N}\omega_{t-1}^{\ell}q(\xi_{t-1}^{\ell},\xi_{t}^{i})}$$

The estimator  $\tilde{\mathcal{T}}_t^N(\xi_t^i)$  is then

$$\tilde{\mathcal{T}}_{t}^{N}(\xi_{t}^{i}) = K^{-1} \sum_{k=1}^{K} \left( \tilde{\mathcal{T}}_{t-1}^{N}(\xi_{t-1}^{J_{t}^{i,k}}) + s_{t-1}(\xi_{t-1}^{J_{t}^{i,k}},\xi_{t}^{i}) \right),$$

and finally the estimator  $\tilde{S}_{t}^{N}$  of  $S_{t}$  is given by

$$\tilde{\mathcal{S}}_t^N = \Omega_t^{-1} \sum_{i=1}^N \omega_t^i \tilde{\mathcal{T}}_t(\xi_t^i).$$

In our novel algorithm we need to draw indices  $J_t^{i,k}$  from the backward kernel, but by applying the same accept-reject-technique as for the FFBSi algorithm described above we obtain an algorithm of computational complexity  $\mathcal{O}(N)$ . Again we require the transition density of the Markov chain to be bounded by some constant  $q_+$ and as in the FFBSi algorithm a candidate backward index is drawn according to the probability distribution proportional to the weights  $\{\omega_t^i\}_{i=1}^N$ . As for the FBBSi, the expected number of draws needed for generating one backward index with this accept-reject scheme can be shown to converge to a constant that does not depend on N; see [11, sec. 2.3].

In our novel algorithm, which is displayed in Algorithm 2, the number K of sub-samples is a design parameter. Increasing K adds computational complexity at the gain of better estimation of  $\mathcal{T}_t$ . In Figure 1 we illustrate the backward sampling procedure in the cases K = 1 and K = 2 when N = 3. In the figure the nodes correspond to particles and the arrows correspond to the indices  $J_t^{i,k}$ drawn in the algorithm for t = 0, ..., 4 and i = 1, 2, 3. From Figure 1 it is clear that the case K = 1 implies a particle path degeneracy phenomenon that is reminiscent with the degeneracy of the genealogical tree that occurs in the standard particle smoother; see [7] [8, Sec 14.3.6] for some discussion. To illustrate further the degeneracy when K = 1 we have estimated the variance of  $S_T = \mathbb{E}\left[\sum_{t=0}^T X_t | Y_{0:T}\right]$  for K = 1 and K = 2. The outcome is displayed in Figure 2 where it is clear that the quadratic trend is present when K = 1 compared to the expected linear trend for this problem. Therefore we conclude that the value of K should be at least 2.

# 5. SIMULATIONS

Simulations comparing the proposed algorithm with the forwardonly version of FFBSm algorithm were implemented on a linear-Gaussian model and a stochastic volatility model. For both models, time averaged sufficient statistics  $\mathcal{S}_t^{(i)}/t$  were computed. Because of the degeneracy phenomenon discussed above we used consequently K = 2 in the simulations. In the results we refer to FoS as the

Algorithm 2 Particle-based, rapid incremental smoother (PaRIS)

- **Require:** Particles and weights  $\{(\xi_{t-1}^i, \omega_{t-1}^i)\}_{i=1}^N$  estimating the filter distribution  $\phi_{t-1}$  and  $\{\tilde{\mathcal{T}}_{t-1}^N(\xi_{t-1}^i)\}_{i=1}^N$ 1: Update the particle sample  $\{(\xi_{t-1}^i, \omega_{t-1}^i)\}_{i=1}^N$  using a particle sample  $\{(\xi_{t-1}^i, \omega_{t-1}^i)\}_{i=1}^N$ 
  - ticle filter of choice in order to obtain a weighted sample  $\{(\xi_t^i, \omega_t^i)\}_{i=1}^N$  approximating  $\phi_t$
- 2: for  $i = 1 \rightarrow N$  do
- 3:
- for  $k = 1 \rightarrow K$  do Simulate  $J_t^{i,k} \sim \{\omega_{t-1}^j q(\xi_{t-1}^j, \xi_t^i)\}_{j=1}^N$  using accept-4: reject sampling.
- 5: end for

6: Set 
$$\tilde{\mathcal{T}}_t^N(\xi_t^i) = K^{-1} \sum_{k=1}^K (\tilde{\mathcal{T}}_{t-1}^N(\xi_{t-1}^{J_t^{(N)}}) + s_{t-1}(\xi_{t-1}^{J_t^{(N)}}, \xi_t^i))$$
  
7: end for

8: Set  $\mathcal{S}_t^{\scriptscriptstyle IN}$  $= \Omega_t^{-1} \sum_{i=1}^N \omega_t^i \mathcal{T}_t^N(\xi_t^i)$ 



Fig. 1. Genealogical traces corresponding to backward simulation in the PaRIS algorithm. Columns of nodes refer to different particle populations (with N = 3) at different time points, with time increasing to the right. Gray-colored nodes are particles that are inactive in the backward approximation after t = 4 time steps.

forward-only version of FFBSm and PaRIS (particle-based, rapid incremental smoother) as our proposed algorithm.

The linear-Gaussian model is of form

$$X_t = aX_{t-1} + \sigma_w W_t,$$
  
$$Y_t = X_t + \sigma_v V_t,$$

where  $\{W_t\}_{t>0}$  and  $\{V_t\}_{t>0}$  are two mutually independent sequences of Gaussian noise. We simulate data using the parameters  $a = 0.8, \sigma_w = 0.2$ , and  $\sigma_v = 1$ . The simulations comprised up to T = 10,000 time points using N = 100 particles for the FoS algorithm and N = 250 particles for the PaRIS algorithm. With these particle sample sizes, our algorithm was still 50% faster than the FoS sampler, due to the quadratic complexity of the latter. For this model we computed the smoothed additive functionals  $S_t^{(i)} = \mathbb{E}[S_t^{(i)}|Y_{0:t}]$ for  $S_t^{(1)} = \sum_{k=0}^{t-1} X_k$ ,  $S_t^{(2)} = \sum_{k=0}^{t-1} X_k X_{k+1}$ . Figure 4 shows boxplots, based on 100 replicates of each algorithm, of the time averaged smoothed additive functionals for different values of t, the stars are the exact values calculated using the Kalman smoother (see e.g. [1, Chap. 5.2.4]). There is some bias present in both algorithms. Calculating the root-mean-square error of these algorithms for the last time point gives us for the FoS algorithm a value of 0.0020 and 0.0013 for  $\mathcal{S}_T^{(1)}/T$  and  $\mathcal{S}_T^{(2)}/T$ , respectively, and using the PaRIS algorithm we get the values 0.0016 and 0.0008, respectively.

To emphasize that just comparing the box-plots dose not provide the full picture we introduce the measure efficiency as the inverse variance over computation time. In Figure 6 the estimated efficiency for  $\mathcal{S}_t^{(1)}$  is plotted for the FoS and PaRIS algorithms, both using N =500 particles. As can be seen the PaRIS algorithm has significantly better efficiency for every time point.

To construct Figure 3 we computed, for all different time points, the average numbers of draws required in the accept-reject sampling



**Fig. 2**. Variance of  $\mathbb{E}\left[\sum_{t=0}^{T} X_t | Y_{0:T}\right]$  for K = 1 and K = 2 fitted to linear and quadratic curves.



Fig. 3. Average numbers of draws needed in the accept-reject sampling scheme at all different time-points for the linear Gaussian model.

scheme per particle and plotted these averages in a histogram. We notice that the total mass of the histogram is contained in the interval (4.9, 6.0).

The stochastic volatility model is given by the state and observation equations

$$X_t = \phi X_{t-1} + \sigma W_t,$$
  
$$Y_t = \beta \exp(X_t/2) V_t.$$

where again  $\{W_t\}_{t\geq 0}$  and  $\{V_t\}_{t\geq 0}$  are two sequences of independent Gaussian noise. We simulated data using the parameters  $\phi = 0.975, \sigma = 0.16$ , and  $\beta = 0.63$ . Again we considered up to T = 10,000 time points, but this time N = 250 particles for both algorithms. With this set-up, our algorithm was 5 times faster than the FoS-implementation. For this model we computed two of the sufficient statistics  $S_t^{(i)} = \mathbb{E}[S_t^{(i)}|Y_{0:t}]$  for  $S_t^{(1)} = \sum_{k=0}^{t-1} X_{k+1}^2$ ,  $S_t^{(2)} = \sum_{k=0}^{t-1} X_k X_{k+1}$ . As it is evident from Figure 5 the performance of our algorithm is, despite being considerably faster, on par with that of the FoS-implementation.

#### 6. DISCUSSION

In this paper we have presented a novel algorithm for efficient forward-only computation of smoothed additive functionals. The proposed algorithm has a computational complexity of  $\mathcal{O}(N)$  which is a drastic improvement compared to existing algorithms having in general  $\mathcal{O}(N^2)$  complexities. The proposed algorithm performs on par with the FoS algorithm but is significantly faster. This coupled



**Fig. 4.** Results of the simulations for linear and Gaussian model. The black stars are the exact values calculated using the Kalman smoother.

with the ease of implementation makes the algorithm highly useful in practice.

For future work the convergence of the algorithm has to be established. The behavior of the algorithm for different values of Kshould also be investigated theoretically, especially the transition from K = 1 to K = 2.



Fig. 5. Results of the simulations for stochastic volatility model.



**Fig. 6**. Estimated efficiencies for estimating  $S_t^{(1)}$  for the linear gaussian model.

### 7. REFERENCES

- [1] Oliver Cappé, Eric Moulines, and Tobias Rydén, *Inference in Hidden Markov Models*, Springer, New York, 2005.
- [2] Pierre Del Moral, Arnaud Doucet, and Sumeetpal S. Singh, "Forward smoothing using sequential Monte Carlo," Tech. Rep., Cambridge University, 2009.
- [3] Oliver Cappé, "Online sequential Monte Carlo EM algorithm," in *Statistical Signal Processing*, 2009. SSP '09. IEEE/SP 15th Workshop on, 2009, pp. 37–40.
- [4] Oliver Cappé, "Online expectation-maximisation," ArXiv eprints, Nov. 2010.
- [5] Christophe Andrieu and Arnaud Doucet, "Online expectationmaximization type algorithms for parameter estimation in general state space models," in *Proc. IEEE Conf. ICASSP*, 2003, pp. 69–72.
- [6] N. J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F. Radar & Signal Processing*, vol. 40, pp. 107–113, 1993.
- [7] Jimmy Olsson, Olivier Cappé, Randal Douc, and Eric Moulines, "Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models," *Bernoulli*, vol. 14, pp. 155–179, 2008.
- [8] Christian P. Robert and George Casella, *Monte Carlo statistical methods. 2nd ed.*, New York, NY: Springer, 2nd ed. edition, 2004.
- [9] Markus Hürzeler and Hans R. Künsch, "Monte Carlo approximations for general state-space models," *Journal of Computational and Graphical Statistics*, vol. 7, no. 2, pp. 175–193, 1998.
- [10] Arnaud Doucet, Simon Godsill, and Christophe Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [11] Randal Douc, Aurélien Garivier, Eric Moulines, and Jimmy Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models," *The Annals of Applied Probability*, vol. 21, pp. 2109–2145, 2011.
- [12] Simon J Godsill, Arnaud Doucet, and Mike West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156– 168, 2004.