Hostettler, Roland; Schön, Thomas B.

Auxiliary-Particle-Filter-Based Two-Filter Smoothing for Wiener State-Space Models

Published in:
Proceedings of the 21st International Conference on Information Fusion, FUSION 2018

DOI:
10.23919/ICIF.2018.8455323

Published: 05/09/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Hostettler, R., & Schön, T. B. (2018). Auxiliary-Particle-Filter-Based Two-Filter Smoothing for Wiener State-Space Models. In *Proceedings of the 21st International Conference on Information Fusion, FUSION 2018* (pp. 1904-1911). Article 8455323 IEEE. https://doi.org/10.23919/ICIF.2018.8455323

Auxiliary-Particle-Filter-Based Two-Filter Smoothing for Wiener State-Space Models

Roland Hostettler and Thomas B. Schön

DOI:

10.23919/ICIF.2018.8455323

Auxiliary-Particle-Filter-based Two-Filter Smoothing for Wiener State-Space Models

Roland Hostettler* and Thomas B. Schön†
*Department of Electrical Engineering and Automation, Aalto University, Finland
E-Mail: roland.hostettler@aalto.fi
†Department of Information Technology, Uppsala University, Sweden
E-Mail: thomas.schon@it.uu.se

***Abstract*—In this paper, we propose an auxiliary-particle-filter-based two-filter smoother for Wiener state-space models. The proposed smoother exploits the model structure in order to obtain an analytical solution for the backward dynamics, which is introduced artificially in other two-filter smoothers. Furthermore, Gaussian approximations to the optimal proposal density and the adjustment multipliers are derived for both the forward and backward filters. The proposed algorithm is evaluated and compared to existing smoothing algorithms in a numerical example where it is shown that it performs similarly to the state of the art in terms of the root mean squared error at lower computational cost for large numbers of particles.**

***Index Terms*—Sequential Monte Carlo, particle filtering, state estimation, state-space models, state-space methods, Wiener models.**

I. Introduction

Sequential Monte Carlo (SMC) methods, including particle filters and smoothers, are a popular approach for solving the Bayesian filtering and smoothing problems arising in nonlinear, non-Gaussian systems that are analytically intractable [1], [2]. In the filtering problem, the objective is to find the posterior distribution of the state $x_n \in \mathbb{R}^{N_x}$ at time t_n (or the state trajectory $x_{1:n} = \{x_1, \dots, x_n\}$ up to t_n) given the observations $y_{1:n} = \{y_1, \dots, y_n\}$ (where $y_n \in \mathbb{R}^{N_y}$), that is, finding $p(x_n \mid y_{1:n})$ (or $p(x_{1:n} \mid y_{1:n})$). Similarly, smoothing is concerned with finding the posterior distribution of the state x_n (or the trajectory $x_{1:N}$) given a batch of data $y_{1:N}$ for $1 \leq n \leq N$, that is, finding $p(x_n \mid y_{1:N})$ (or $p(x_{1:N} \mid y_{1:N})$) [2], [3].

Particle filters and smoothers have successfully been employed in a variety of applications, see, for example [4]–[8]. Nevertheless, one of the main challenges of these methods is that they scale poorly in terms of dimensionality, meaning that systems with large state dimension N_x require a large number of Monte Carlo samples M (referred to as the curse of dimensionality) [9]. This is particularly problematic for smoothing, which often scales worse with respect to M compared to filtering. Approaches to mitigate this issue include the construction of better proposals that manage to keep the particles in the relevant area of the state-space [10]–[13] or exploiting the model structure, for example, by using Rao–Blackwellization in conditionally linear Gaussian state-space models [14]–[16].

Another class of models where we can exploit the structure are Wiener state-space models [17], consisting of linear, Gaussian dynamics and nonlinear observations of the form

$$x_n = A_n x_{n-1} + \epsilon_n, \tag{1a}$$
$$y_n = g_n(x_n, r_n), \tag{1b}$$
$$x_0 \sim \mathcal{N}(\mu_0^x, \Sigma_0^x), \tag{1c}$$

where A_n is the state transition matrix, $\epsilon_n \sim \mathcal{N}(0, Q_n)$ is the process noise and $\mathcal{N}(\mu, \Sigma)$ denotes the multivariate Gaussian density with mean μ and covariance Σ, $g_n(\cdot)$ is the nonlinear observation function, and $r_n \sim p(r_n)$ is measurement noise.

In this paper, we will show how to exploit the structure in the model (1) in order to obtain analytical expressions for the backward dynamics, which yields a two-filter smoother [18] that does not rely on artificially introduced backward dynamics as required by the currently known two-filter smoothers [19], [20]. The proposed smoother uses forward and backward filters that are based on the auxiliary particle filter (APF) [10] which has the advantageous property of concentrating the particles in areas of high probability. Since the measurement model is nonlinear, the optimal proposals are analytically intractable and we instead propose to use Gaussian approximations using moment matching [11], [21], [22]. The main advantages of the proposed method are: 1) The backward filter exploits the structure inherent in the model to obtain a suitable importance distribution; and 2) the computational complexity is reduced thanks to efficient implementation. The method presented here is an extension of the earlier work in [23]. The main improvements are that we develop a more robust algorithm based on the (approximately) fully adapted auxiliary particle filter. Furthermore, both the forward and backward filtering stages are discussed in more detail, a complexity analysis is included, and more thorough numerical illustrations and comparisons are provided.

II. Related Work

The most basic smoother is obtained by running a particle filter targeting the joint filtering distribution over the complete data $y_{1:N}$ [24]. This approximation of the smoothing density is of limited use, since the particle filter produces degenerate state trajectories. This means that for long time series, all particles at t_N will have one (or at best a few) common ancestor(s) [2], [24], [25]. However, the marginal filtering

approximations obtained from a particle filter can be used in a backward pass similar to the Rauch–Tung–Striebel smoother for linear systems [26]. This yields the forward filtering backward smoothing method which reweighs the filtered particles in the backward pass. Unfortunately, the resulting algorithm scales poorly, namely according to $\mathcal{O}(M^2N)$, which renders it prohibitive for anything but low numbers of particles M [2].

The forward filtering backward simulation (FFBSi) smoother [2], [27] is based on first running a particle filter forward in time, which generates a set of M_F (degenerate) weighted forward trajectories. This is followed by a simulation pass backwards in time where a partial trajectory $x^j_{n+1:N}$ is extended with a new sample x^m_n from the forward filtering approximation at time t_n such that $x^j_{n:N} = \{x^m_n, x^j_{n+1:N}\}$. Extending the trajectory is achieved by sampling from a categorical distribution where the weights for x^m_n are proportional to $p(x^j_{n+1} \mid x^m_n)$. This requires evaluation of $p(x^j_{n+1} \mid x^m_n)$ for each $m \in 1, \ldots, M_F$. Thus, when generating M_S backward trajectories, this requires $\mathcal{O}(M_S M_F)$ evaluations at each time step t_n. However, noting that only one particle is used for extending each trajectory, rejection sampling can be used to replace direct sampling from the categorical distribution [28]. In theory, this greatly alleviates the computational burden, however, in practice, it might suffer from high rejection rates. If this is the case, a solution is to fall back to the original exhaustive search [29]. Further approaches to relieve the computational burden in FFBSi include Rao–Blackwellization of analytically tractable substructures [16] or combining the FFBSi smoother with Markov Chain Monte Carlo (MCMC) moves in the backward simulation pass [30], [31].

Another approach to smoothing is based on a two-filter formulation [32], [33]. Here, the smoothing density is factorized in a way that makes it possible to decouple the smoothing problem into two filters, a regular filter processing the data forward in time and an information filter running backward in time, followed by a combination of the two filters which finally yields the smoothing density approximation. A problem here is that due to the nonlinearity in the process model, carefully selected artificial importance densities have to be introduced for the backward filter [19], [20], [22], [24], [34]–[36].

The algorithm by Kronander et. al. [37] uses a third way of factorizing the marginal smoothing density. Similarly to the FFBSi smoother, a particle filter is run in the forward direction and the particles produced by that filter are resampled in a backward pass. However, unlike the FFBSi smoother, the marginal smoothing distribution is approximated using a weighted set of particles where the weight is proportional to the transition density between the particle at time t_n and an ancestor particle at t_{n+1}. This yields a fast smoother, which, however, suffers from a slight bias under certain conditions [37].

Particle MCMC smoothers differ quite significantly from the smoothers discussed so far [38]. In these approaches, a Gibbs or Metropolis–Hastings sampler is constructed that essentially produces complete trajectory samples $x_{1:N}$ from the joint smoothing distribution $p(x_{1:N} \mid y_{1:N})$. MCMC smoothers based on the Gibbs sampler use a conditional particle filter to sample the trajectories, which ensures that the resulting Markov chain is invariant [38]. To improve convergence, mixing inside the conditional particle filter, and hence, reduce burn-in, further modifications such as ancestor sampling have been introduced [31], [39]–[42].

Finally, an online approach to smoothing was proposed in [43], [44]. In contrast to the methods discussed so far, this approach does not approximate the smoothing density. Instead, it estimates smoothed expectations of functionals of the state using SMC. The resulting algorithm has linear computational complexity and can run online since it does not require processing the data in the reverse temporal direction.

III. Auxiliary Particle Filter

In this section, the auxiliary particle filter (APF) [10] is reviewed to lay the foundation for the coming derivations.

The APF introduces an auxiliary variable that makes the stochastic nature of the resampling step in sequential importance resampling explicit. Assume that we are given the particle approximation of the joint filtering distribution $p(x_{1:n-1} \mid y_{1:n-1})$ at time t_{n-1}

$$\begin{aligned} p(x_{1:n-1} \mid y_{1:n-1}) &\approx \sum_{m=1}^{M} w^m_{n-1} \delta(x_{1:n-1} - x^m_{1:n-1}) \\ &\triangleq \hat{p}(x_{1:n-1} \mid y_{1:n-1}), \end{aligned} \tag{2}$$

where $x^m_{1:n-1}$ and w^m_{n-1} denote the mth particle and its weight, $\delta(\cdot)$ is the Dirac delta function, and $\hat{p}(\cdot)$ is the particle approximation of the density $p(\cdot)$. The marginal filtering density is readily obtained by marginalizing (2) with respect to $x_{1:n-2}$. Hence, for simplicity, we will work with the joint filtering density for the remainder of this work.

During resampling, when going from time t_{n-1} to time t_n, the particles are resampled such that

$$\Pr\{\bar{x}^i_{n-1} = x^m_{n-1}\} = w^m_{n-1},$$

where $\bar{x}^i_{n-1}$ denotes the resampled particle. The randomness of this step is made explicit by introducing the auxiliary variable (referred to as the ancestor index) α_n drawn from the categorical distribution of the weights $\mathcal{C}(\{w^m_{n-1}\}_{m=1}^M)$

$$\alpha^i_n \sim \mathcal{C}(\{w^m_{n-1}\}_{m=1}^M)$$

and letting the resampled particle

$$\bar{x}^i_{n-1} \leftarrow x^{\alpha^i_n}_{n-1}.$$

However, in the APF we actually go one step further and consider a more generic proposal distribution for α_n that also takes the latest measurement y_n into account in order to resample the most likely particles. To this end, so-called adjustment multipliers $f(x_{n-1}, y_n)$ are introduced such that

$$v^m_n \propto w^m_{n-1} f(x^m_{n-1}, y_n) \tag{3}$$

and then draw the auxiliary variables from the categorical distribution of the modified weights $\mathcal{C}(\{v^m_n\}_{m=1}^M)$ as

$$\alpha^i_n \sim \mathcal{C}(\{v^m_n\}_{m=1}^M). \tag{4}$$

Algorithm 1 Auxiliary Particle Filter (APF)

1: Sample $x_0^m \sim p(x_0)$ and set $w_0^m = 1/M$
2: **for** n = 1, ... **do**
3: Sample $\alpha_n^m \sim q(\alpha_n \mid x_{n-1}^m, y_n)$
4: Sample $x_n^m \sim q(x_n \mid \alpha_n^m, x_{n-1}^m, y_n)$
5: Calculate and normalize the importance weights

$$\tilde{w}^m = w_{n-1}^m \frac{p(y_n \mid x_n^m)p(x_n^m \mid x_{n-1}^m)}{q(x_n^m \mid \alpha_n^m, x_{n-1}^m, y_n)q(\alpha_n^m \mid x_{n-1}^m, y_n)}$$
$$w_n^m = \frac{\tilde{w}^m}{\sum_{k=1}^M \tilde{w}^k}$$

6: **end for**

Thus, when targeting the joint filtering density

$$p(x_{1:n} \mid y_{1:n}) \propto p(y_n \mid x_n)p(x_n \mid x_{n-1})p(x_{1:n-1} \mid y_{1:n-1})$$

we search for a joint importance density that factorizes as

$$\begin{aligned} &q(x_{1:n}, \alpha_{1:n} \mid y_{1:n}) \\ &\quad = q(x_n, \alpha_n \mid x_{n-1}, y_n)q(x_{1:n-1}, \alpha_{1:n-1} \mid y_{1:n-1}) \\ &\quad = q(x_n \mid \alpha_n, x_{n-1}, y_n)q(\alpha_n \mid x_{n-1}, y_n) \\ &\qquad \times q(x_{1:n-1}, \alpha_{1:n-1} \mid y_{1:n-1}), \end{aligned} \tag{5}$$

with $q(\alpha_n \mid x_{n-1}, y_n)$ according to (3)–(4). Hence, the importance weights become

$$\begin{aligned} w_n^m &= \frac{p(x_{1:n}^m \mid y_{1:n})}{q(x_{1:n}^m, \alpha_{1:n}^m \mid y_{1:n})} \\ &= w_{n-1}^m \frac{p(y_n \mid x_n^m)p(x_n^m \mid x_{n-1}^m)}{v_n^m q(x_n^m \mid \alpha_n^m, x_{n-1}^m, y_n)}. \end{aligned} \tag{6}$$

It is well known that the optimal (in the sense of minimum weight increment variance) choices for the proposal $q(x_n \mid \alpha_n, x_{n-1}, y_n)$ and the adjustment multipliers are [10], [45]

$$q(x_n^m \mid \alpha_n^m, x_{n-1}^m, y_n) = p(x_n^m \mid x_{n-1}^{\alpha_n^m}, y_n), \tag{7a}$$
$$f(x_{n-1}^m, y_n) = p(y_n \mid x_{n-1}^m), \tag{7b}$$

which reduce the importance weights to $1/M$. This yields the fully adapted APF summarized in Algorithm 1.

IV. Filtering

Having introduced the general APF in Section III, this section shows how the filter can be applied to models of the form (1) to target the joint filtering density $p(x_{1:n} \mid y_{1:n})$. In particular, suitable approximations of the optimal proposal and the adjustment multipliers are derived using the general framework of moment matching, which can be solved using techniques such as linearization or sigma-point integration [46], [47]. Much of the developments in this section will be reused in deriving the backward filter and the smoother in the following section. In this section, the subscript $n \mid 1:n$ is used for particles and their weights in order to indicate the forward filtering particle system while the subscripts $n \mid n-1$ and $n \mid n$ are used to indicate conditioning on x_{n-1} and y_n, respectively.

It can be seen from (1) that closed-form expressions for the optimal proposal and adjustment multipliers can not be derived for state-space models of the Wiener-type. Thus, we propose to approximate the joint distribution $p(x_n, y_n \mid x_{n-1})$ as a Gaussian distribution and use this as an approximation of the optimal proposal distribution. This will exploit the fact that the state dynamics are linear and Gaussian, see (1a). Note, however, that other approximations such as, for example Gaussian mixtures [48], are possible and might be better suited for certain problems (e.g. when the likelihood is multi-modal).

In order to develop the approximate proposal distribution, consider the following Gaussian approximation of the joint density of x_n and y_n given x_{n-1}

$$\begin{aligned} &p(x_n, y_n \mid x_{n-1}) \\ &\quad \approx \mathcal{N}\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}; \begin{bmatrix} A_n x_{n-1} \\ \mu_{n|n-1}^y \end{bmatrix}, \begin{bmatrix} Q_n & B_{n|n-1} \\ B_{n|n-1}^{\mathsf{T}} & S_{n|n-1} \end{bmatrix}\right) \\ &\quad \triangleq \tilde{p}(x_n, y_n \mid x_{n-1}), \end{aligned} \tag{8}$$

where $\tilde{p}(\cdot)$ denotes the Gaussian approximation of the density $p(\cdot)$. The mean $\mu_{n|n-1}^y$ and the covariances $B_{n|n-1}$ and $S_{n|n-1}$ can be found through moment matching [3], [21] using

$$\mu_{n|n-1}^y = \mathrm{E}\{y_n\}, \tag{9a}$$
$$B_{n|n-1} = \mathrm{E}\{(x_n - A_n x_{n-1})(y_n - \mu_{n|n-1}^y)^{\mathsf{T}}\}, \tag{9b}$$
$$S_{n|n-1} = \mathrm{E}\{(y_n - \mu_{n|n-1}^y)(y_n - \mu_{n|n-1}^y)^{\mathsf{T}}\}, \tag{9c}$$

where the expectations are with respect to $p(x_n, r_n \mid x_{n-1}) = p(x_n \mid x_{n-1})p(r_n)$ and $p(r_n)$ denotes the density of the measurement noise as defined in (1).

Conditioning the approximation in (8) on y_n yields [3]

$$\begin{aligned} \tilde{p}(x_n \mid x_{n-1}, y_n) &= \mathcal{N}(x_n; \mu_{n|n}^x, C_{n|n}^x) \\ &\triangleq q(x_n \mid x_{n-1}, y_n), \end{aligned} \tag{10}$$

with

$$\mu_{n|n}^x = A_n x_{n-1} + B_{n|n-1} S_{n|n-1}^{-1}(y_n - \mu_{n|n-1}^y), \tag{11a}$$
$$C_{n|n}^x = Q_n - B_{n|n-1} S_{n|n-1}^{-1} B_{n|n-1}^{\mathsf{T}}. \tag{11b}$$

Similarly, an approximation for $p(y_n \mid x_{n-1})$ is found by simply marginalizing (8) with respect to x_n which yields

$$\tilde{p}(y_n \mid x_{n-1}) = \mathcal{N}(y_n; \mu_{n|n-1}^y, S_{n|n-1}). \tag{12}$$

Using these approximations, the particle weights (6) become

$$\begin{aligned} w_{n|1:n} &\propto w_{n-1|1:n-1} \frac{p(y_n \mid x_n)p(x_n \mid x_{n-1})}{v_{n-1|1:n-1} q(x_n \mid x_{n-1}, y_n)} \\ &\propto w_{n-1|1:n-1} \frac{p(y_n \mid x_n)p(x_n \mid x_{n-1})}{w_{n-1|1:n-1}\tilde{p}(y_n \mid x_{n-1})} \\ &\qquad \times \frac{\tilde{p}(y_n \mid x_{n-1})}{\tilde{p}(y_n \mid x_n, x_{n-1})\tilde{p}(x_n \mid x_{n-1})} \\ &= \frac{p(y_n \mid x_n)}{\tilde{p}(y_n \mid x_n, x_{n-1})}, \end{aligned} \tag{13}$$

where the last equality is due to the fact that $\tilde{p}(x_n \mid x_{n-1})$ is exact. The density $\tilde{p}(y_n \mid x_n, x_{n-1})$ is obtained by conditioning (8) on x_n and it is given by

$$\tilde{p}(y_n \mid x_n, x_{n-1}) = \mathcal{N}(y_n; \mu_{n|n-1:n}^y, C_{n|n-1:n}^y), \tag{14}$$

Algorithm 2 Forward Filter Iteration

1: Calculate the moments $\mu^{y,m}_{n|n-1}$, $B^m_{n|n-1}$, and $S^m_{n|n-1}$ according to (9) with $x_{n-1} = x^m_{n-1}$
2: Calculate and normalize the auxiliary variable probabilities

$$\bar{v}^m = w^m_{n-1|n-1}\mathcal{N}(y_n; \mu^{y,m}_{n|n-1}, S^m_{n|n-1})$$
$$v^m_{n|n} = \frac{\bar{v}^m}{\sum_{k=1}^M \bar{v}^k}$$

3: Sample $\alpha^m_n \sim \mathcal{C}(\{v^k_{n|n}\}_{k=1}^M)$
4: Calculate $\mu^{x,m}_{n|n}$ and $C^m_{n|n}$ according to (11)
5: Sample $x^m_n \sim \mathcal{N}(\mu^{x,\alpha^m_n}_{n|n}, C^{\alpha^m_n}_{n|n})$
6: Calculate $\mu^{y,m}_{n|n-1:n}$ and $C^{y,m}_{n|n-1:n}$ according to (15) using $\mu^{y,\alpha^m}_{n|n-1}$, $B^{\alpha^m}_{n|n-1}$, x^m_n, and $x^{\alpha^m}_{n-1}$
7: Calculate and normalize the particle weights

$$\bar{w}^m = \frac{p(y_n \mid x^m_n)}{\mathcal{N}(y_n; \mu^{y,m}_{n|n-1:n}, C^{y,m}_{n|n-1:n})}$$
$$w^m_{n|n} = \frac{\bar{w}^m}{\sum_{k=1}^M \bar{w}^k}$$

with

$$\mu^y_{n|n-1:n} = \mu^y_{n|n-1} + B^{\mathsf{T}}_{n|n-1}Q_n^{-1}(x_n - A_n x_{n-1}), \quad (15a)$$
$$C^y_{n|n-1:n} = S_{n|n-1} - B^{\mathsf{T}}_{n|n-1}Q_n^{-1}B_{n|n-1}. \quad (15b)$$

Finally, one iteration of the complete approximate fully adapted APF is summarized in Algorithm 2. Equations (10)–(12) and (14) provide the approximations of the distributions required by the APF for each particle. Thus, the mean $\mu^y_{n|n-1}$ as well as the covariance matrices $B_{n|n-1}$ and $S_{n|n-1}$ have to be calculated using (9) for every particle x^m_{n-1} $(m = 1, \ldots, M)$.

To use Gaussian approximations for the optimal importance density and adjustment multipliers is a standard approach in particle filtering [11], [22]. A challenge is that in most cases, the integrals (9) can not be evaluated in closed-form. Instead, they need to be approximated using well-known techniques commonly used in Kalman filtering such as sigma-point methods (unscented transform, spherical cubature, Gauss–Hermite quadrature, etc.) or linearization, see [3], [11], [21].

V. Smoothing

In this section, the two-filter formulation is first reviewed and then, a backward filter similar to the forward filter is developed. Finally, the two filters are combined to obtain the smoother.

A. Two-Filter Smoothing

The general two-filter smoothing formulation is [18], [24]

$$p(x_n \mid y_{1:N}) \propto p(x_n \mid y_{1:n-1})p(y_{n:N} \mid x_n), \quad (16)$$

where the predictive density can readily be obtained from a forward filter. The second term, $p(y_{n:N} \mid x_n)$, is unfortunately not a probability density in x_n. However, it can be further reformulated using Bayes' rule, resulting in [18], [23]

$$\begin{aligned} p(y_{n:N} \mid x_n) &\propto \frac{p(x_n \mid y_{n:N})}{p(x_n)} \\ &= \frac{1}{p(x_n)} \int p(x_{n:N} \mid y_{n:N}) \mathrm{d}x_{n+1:N}, \end{aligned} \quad (17)$$

where $p(x_n \mid y_{n:N})$ is the backward filtering density which can be seen as the marginal of the joint backward filtering density $p(x_{n:N} \mid y_{n:N})$. The latter can be factorized such that a recursive formulation similar to the forward filter is obtained:

$$\begin{aligned} p(x_{n:N} \mid y_{n:N}) \propto\ & p(y_n \mid x_n)p(x_n \mid x_{n+1}) \\ & \times p(x_{n+1:N} \mid y_{n+1:N}). \end{aligned} \quad (18)$$

For arbitrary nonlinear, non-Gaussian systems, the densities $p(x_n)$ and $p(x_n \mid x_{n+1})$ can not be calculated in closed form, which makes this type of smoother difficult to implement. One approach is to introduce artificial densities for $p(x_n)$ and $p(x_n \mid x_{n+1})$ in order to obtain the formulation (17)–(18) [19], [20]. However, for the Wiener state-space model (1), the linearity in the state dynamics allows us to find closed-form solutions for $p(x_n)$ and $p(x_n \mid x_{n+1})$, which is the property exploited in this work.

B. Backward Filter

Based on the joint backward filtering density (18), a backward particle filter can now be developed with (18) as the target density. Similar to the forward filter, we will make use of an importance distribution that factorizes according to

$$\begin{aligned} &q(x_{n:N}, \beta_{n:N} \mid y_{n:N}) \\ &\quad = q(x_n \mid \beta_n, x_{n+1:N}, y_{n:N})q(\beta_n \mid x_{n+1:N}, y_{n:N}) \\ &\qquad \times q(x_{n+1:N}, \beta_{n+1:N} \mid y_{n+1:N}), \end{aligned}$$

where β_n are the auxiliary variables. Then, the weights are

$$\begin{aligned} w_{n|n:N} = w_{n+1|n+1:N} &\frac{p(y_n \mid x_n)p(x_n \mid x_{n+1})}{q(\beta_n \mid x_{n+1:N}, y_{n:N})} \\ &\times \frac{1}{q(x_n \mid \beta_n, x_{n+1:N}, y_{n:N})}. \end{aligned} \quad (19)$$

It can be shown that the proposal for x_n minimizing the variance of (19) is, analogously to the forward filter, given by

$$q(x_n \mid x_{n+1:N}, y_{n:N}) = p(x_n \mid x_{n+1}, y_n) \quad (20)$$

and then, the optimal adjustment multipliers are given by $p(y_n \mid x_{n+1})$ (the proof follows the same lines as the proof for the forward filter, see [2], [45] for details). Unfortunately, this proposal is again unavailable in general but a Gaussian approximation as in Section IV can be used instead (among other choices). First, note that $p(x_n \mid x_{n+1})$ can be calculated exactly as follows. Consider the joint density of x_n and x_{n+1}

$$\begin{aligned} &p(x_n, x_{n+1}) \\ &\quad = \mathcal{N}\left(\begin{bmatrix} x_n \\ x_{n+1} \end{bmatrix}; \begin{bmatrix} \mu^x_n \\ \mu^x_{n+1} \end{bmatrix}, \begin{bmatrix} \Sigma^x_n & \Sigma^x_n A^{\mathsf{T}}_{n+1} \\ A_{n+1}\Sigma^x_n & \Sigma^x_{n+1} \end{bmatrix}\right), \end{aligned}$$

where μ_n^x and Σ_n^x are computed recursively according to

$$\mu_n^x = A_n \mu_{n-1}^x, \tag{21a}$$
$$\Sigma_n^x = A_n \Sigma_{n-1}^x A_n^{\mathsf{T}} + Q_n. \tag{21b}$$

Conditioning on x_{n+1} yields

$$p(x_n \mid x_{n+1}) = \mathcal{N}(x_n; \mu_{n|n+1}^x, \Sigma_{n|n+1}^x), \tag{22}$$

with

$$\mu_{n|n+1}^x = \mu_n^x + \Sigma_n^x A_{n+1}^{\mathsf{T}} (\Sigma_{n+1}^x)^{-1} (x_{n+1} - \mu_{n+1}^x), \tag{23a}$$
$$\Sigma_{n|n+1}^x = \Sigma_n^x - \Sigma_n^x A_{n+1}^{\mathsf{T}} (\Sigma_{n+1}^x)^{-1} A_{n+1} \Sigma_n^x. \tag{23b}$$

Next, the joint Gaussian approximation of x_n and y_n given by

$$\begin{aligned} &p(x_n, y_n \mid x_{n+1}) \\ &\quad \approx \mathcal{N}\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}; \begin{bmatrix} \mu_{n|n+1}^x \\ \mu_{n|n+1}^y \end{bmatrix}, \begin{bmatrix} \Sigma_{n|n+1}^x & B_{n|n+1} \\ B_{n|n+1}^{\mathsf{T}} & S_{n|n+1} \end{bmatrix} \right) \\ &\quad \triangleq \tilde{p}(x_n, y_n \mid x_{n+1}), \end{aligned} \tag{24}$$

is introduced, where $\mu_{n|n+1}^y$, $B_{n|n+1}$, and $S_{n|n+1}$ can be found through moment matching with respect to $p(x_n, r_n \mid x_{n+1}) = p(x_n \mid x_{n+1}) p(r_n)$. The corresponding expectations are

$$\mu_{n|n+1}^y = \mathrm{E}\{y_n\}, \tag{25a}$$
$$B_{n|n+1} = \mathrm{E}\{(x_n - \mu_{n|n+1}^x)(y_n - \mu_{n|n+1}^y)^{\mathsf{T}}\}, \tag{25b}$$
$$S_{n|n+1} = \mathrm{E}\{(y_n - \mu_{n|n+1}^y)(y_n - \mu_{n|n+1}^y)^{\mathsf{T}}\}. \tag{25c}$$

Conditioning (24) on y_n then yields

$$\begin{aligned} \tilde{p}(x_n \mid x_{n+1}, y_n) &= \mathcal{N}(x_n; \mu_{n|N}^x, C_{n|N}) \\ &\triangleq q(x_n \mid x_{n+1}, y_n), \end{aligned} \tag{26}$$

with

$$\mu_{n|n:n+1}^x = \mu_{n|n+1}^x + B_{n|n+1} S_{n|n+1}^{-1} (y_n - \mu_{n|n+1}^y), \tag{27a}$$
$$C_{n|n:n+1}^x = \Sigma_{n|n+1}^x - B_{n|n+1} S_{n|n+1}^{-1} B_{n|n+1}^{\mathsf{T}}. \tag{27b}$$

The approximate adjustment multipliers are readily obtained from (24) by marginalizing with respect to x_n and they are

$$\tilde{p}(y_n \mid x_{n+1}) = \mathcal{N}(y_n; \mu_{n|n+1}^y, S_{n|n+1}). \tag{28}$$

The unnormalized backward filter weights thus become

$$w_{n|n:N} \propto \frac{p(y_n \mid x_n)}{\tilde{p}(y_n \mid x_n, x_{n+1})}, \tag{29}$$

with

$$\tilde{p}(y_n \mid x_n, x_{n+1}) = \mathcal{N}(y_n; \mu_{n|n:n+1}^y, C_{n|n:n+1}^y), \tag{30}$$

and

$$\begin{aligned} \mu_{n|n:n+1}^y &= \mu_{n|n+1}^y \\ &\quad + B_{n|n+1}^{\mathsf{T}} (\Sigma_{n|n+1}^x)^{-1} (x_n - \mu_{n|n+1}^x), \end{aligned} \tag{31a}$$
$$C_{n|n:n+1}^y = S_{n|n+1} - B_{n|n+1}^{\mathsf{T}} (\Sigma_{n|n+1}^x)^{-1} B_{n|n+1}. \tag{31b}$$

Algorithm 3 Backward Filter Iteration

1: Calculate $\mu_{n|n+1}^{x,m}$ and $\Sigma_{n|n+1}^{x,m}$ according to (23) using $x_{n+1|n+1:N}^m$
2: Calculate the moments $\mu_{n|n+1}^{y,m}$, $B_{n|n+1}^m$, and $S_{n|n+1}^m$ according to (25) using $x_{n+1|n+1:N}^m$
3: Calculate and normalize the auxiliary variable probabilities

$$\bar{v}^m = w_{n+1|n+1:N}^m \mathcal{N}(y_n; \mu_{n|n+1}^{y,m}, S_{n|n+1}^m)$$
$$v_{n|n:N}^m = \frac{\bar{v}^m}{\sum_{k=1}^M \bar{v}^k}$$

4: Sample $\beta_n^m \sim \mathcal{C}(\{v_{n|n:N}^k\}_{k=1}^M)$
5: Calculate $\mu_{n|n:n+1}^{x,\beta_n^m}$ and $C_{n|n:n+1}^{x,\beta_n^m}$ according to (27) using $x_{n+1|n+1:N}^{\beta_n^m}$
6: Sample $x_{n|n:N}^m \sim \mathcal{N}(\mu_{n|n:n+1}^{x,\beta_n^m}, C_{n|n:n+1}^{x,\beta_n^m})$
7: Calculate $\mu_{n|n:n+1}^{y,m}$ and $C_{n|n:n+1}^{y,m}$ according to (31) using $\mu_{n|n+1}^{y,\beta^m}$, $B_{n|n+1}^{\beta^m}$, $x_{n|n:N}^m$, and $\mu_{n|n+1}^{x,\beta^m}$
8: Calculate and normalize particle weights

$$\bar{w}^m = \frac{p(y_n \mid x_n^m)}{\mathcal{N}(y_n; \mu_{n|n:n+1}^{y,m}, C_{n|n:n+1}^{y,m})}$$
$$w_{n|n:N}^m = \frac{\bar{w}^m}{\sum_{k=1}^M \bar{w}^k}$$

One iteration of the backward filter can now be summarized and is given in Algorithm 3, where $x_{n|n:N}^m$ denotes the mth particle in the backward direction.

It remains to find an initialization for the backward filter at $n = N$. First note that at that time, the target density is

$$p(x_N \mid y_N) \propto p(y_N \mid x_N) p(x_N). \tag{32}$$

A straightforward approach is then to reuse the particles from the forward filter and sample from

$$\hat{p}(x_N \mid y_{1:N}) \approx \sum_{m=1}^M w_{N|N}^m \delta(x_N - x_{N|N}^m), \tag{33}$$

where the filtered weights at time N are given by

$$w_{N|N} \propto \frac{p(y_N \mid x_N)}{\tilde{p}(y_N \mid x_N, x_{N-1})}.$$

Thus, choosing

$$v_{N|N} \propto p(x_N) \tilde{p}(y_N \mid x_N, x_{N-1}) \tag{34}$$

and sampling according to

$$q(\beta_N \mid y_N) = \mathcal{C}(\{v_{N|N}^m\}_{m=1}^M), \tag{35a}$$
$$q(x_N \mid \beta_N, y_N) = \hat{p}(x_N \mid y_{1:N}), \tag{35b}$$

yields that the smoothed weights become

$$w_{N|N} \propto \frac{p(y_N \mid x_N) p(x_N)}{q(\beta_N \mid y_N) q(x_N \mid \beta_N, y_N)} = 1. \tag{36}$$

Algorithm 4 Smoother

1: Sample $x^m_{0|0} \sim \mathcal{N}(\mu^x_0, \Sigma^x_0)$ and set $w^m_{0|0} = 1/M$.
2: **for** $n = 1, \ldots, N$ **do**
3: Run the forward filter iteration in Algorithm 2.
4: Calculate μ^x_n and Σ^x_n according to (21).
5: **end for**
6: Initialize $x^m_{N|N}$ and $w^m_{N|N}$ according to (35) and (36).
7: **for** $n = N-1, \ldots, 1$ **do**
8: Run the backward filter iteration in Algorithm 3.
9: Calculate and normalize the smoothed particle weights according to (39).
10: **end for**

C. Resulting Smoother

Having developed both the forward and backward filters, the resulting smoother can now be assembled. First, note that

$$\hat{p}(y_{n:N} \mid x_n) \propto \sum_{m=1}^{M} \frac{w^m_{n|n:N}}{p(x^m_{n|n:N})} \delta(x_n - x^m_{n|n:N}), \tag{37}$$

which is obtained by replacing the backward filtering density by its particle approximation in (17). Next, using (37) together with the particle approximation of the predictive density

$$\hat{p}(x_n \mid y_{1:n-1}) = \sum_{k=1}^{M} w^k_{n-1|1:n-1} p(x_n \mid x^k_{n-1|1:n-1})$$

in (16) gives

$$\begin{aligned} p(x_n \mid y_{1:N}) &\propto \sum_{k=1}^{M} w^k_{n-1|1:n-1} p(x_n \mid x^k_{n-1|1:n-1}) \\ &\quad \times \sum_{m=1}^{M} \frac{w^m_{n|n:N}}{p(x^m_{n|n:N})} \delta(x_n - x^m_{n|n:N}) \\ &= \sum_{m=1}^{M} \sum_{k=1}^{M} w^k_{n-1|1:n-1} p(x^m_{n|n:N} \mid x^k_{n-1|1:n-1}) \\ &\quad \times \frac{w^m_{n|n:N}}{p(x^m_{n|n:N})} \delta(x_n - x^m_{n|n:N}). \end{aligned} \tag{38}$$

Thus, we can define the smoothed particle weights as

$$\begin{aligned} w^m_{n|1:N} &\propto \frac{w^m_{n|n:N}}{p(x^m_{n|n:N})} \\ &\quad \times \sum_{k=1}^{M} w^k_{n-1|1:n-1} p(x^m_{n|n:N} \mid x^k_{n-1|1:n-1}) \end{aligned} \tag{39}$$

and obtain the particle approximation

$$\hat{p}(x_n \mid y_{1:N}) = \sum_{m=1}^{M} w^m_{n|1:N} \delta(x_n - x^m_{n|n:N}) \tag{40}$$

for the marginal smoothing density. This finally yields the complete smoothing algorithm as listed in Algorithm 4.

D. Computational Complexity

The proposed smoother (Algorithm 4) consists of three main steps: 1) Forward filtering, 2) backward filtering, and 3) calculation of the smoothed weights.

Since both the forward filter and the backward filter are based on the APF, it follows that their computational complexity is of order $\mathcal{O}(M)$. Also, all steps except for the weight normalization can be parallelized. Calculating the smoothed weights scales quadratically in M due to the double sum in (38). However, since the transition density is linear and Gaussian, this can be computed efficiently by evaluating $p(x^m_{n|n:N} \mid x^k_{n-1|1:n-1})$ for all k at once. Hence, the overall complexity of the smoother is of order $\mathcal{O}(M^2)$ with a low constant.

Note that the algorithm can be sped up by using the bootstrap proposal in both the forward and backward directions, rather than the approximations of the optimal proposals. In that case, calculation of the linearized moments, which has to be done for each particle individually, is not required and extending the state trajectories can be done very efficiently (at the expense of a less accurate proposal), see [23].

VI. Numerical Illustrations

In this section, we illustrate the performance of the proposed method in an example, where we also compare the method to the existing methods from the literature. In particular, it is compared to the forward filtering backward simulation (FFBSi) particle smoother [27], the marginal smoother from [37] (KSD), as well as a particle Gibbs with ancestor sampling-based MCMC smoother (CPF-AS) [41], [42].

A. Setup

The system under consideration is given by

$$\begin{aligned} x_n &= \begin{bmatrix} -0.368 & -0.888 & -0.524 & -0.555 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_{n-1} + \epsilon_n, \\ z_n &= \begin{bmatrix} 1 & 0.1 & -0.49 & 0.01 \end{bmatrix} x_n, \\ y_n &= g(z_n) + r_n, \end{aligned}$$

with

$$g_n(z_n) = \begin{cases} 1 + z & z < -1, \\ -\frac{\sin(\pi z)}{\pi} & -1 \leq z \leq 1, \\ -1 + z & z > 1, \end{cases}$$

which is the non-monotonic model that was also considered in [49]. Furthermore, μ^x_0, Σ^x_0, Q, and R were chosen as

$$\mu^x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}, \Sigma^x_0 = I_4, Q = 0.25^2 I_4, R = 0.1.$$

We evaluate the smoothers for $M = 100, 200, \ldots, 1000$ particles. For the proposed method, we use the same number of particles in the forward and backward filters, while for the FFBSi smoother, we simulate $M/2$ backward trajectories. For the CPF-AS, we use the same number of particles in the individual particle filters and show the results for $K = 10$ and $K = 20$ trajectories drawn from the posterior. The filter

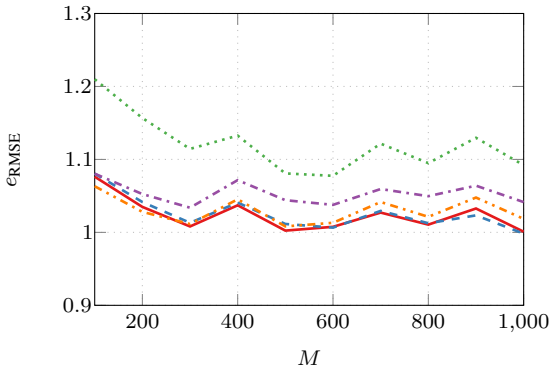

Fig. 1. Comparison of the time-averaged RMSE e_{RMSE} for the proposed smoother (——), FFBSi (– – –), KSD (·····), CPF-AS with $K = 10$ trajectories (–··–··–), and CPF-AS with $K = 20$ trajectories (–·––·––·–).

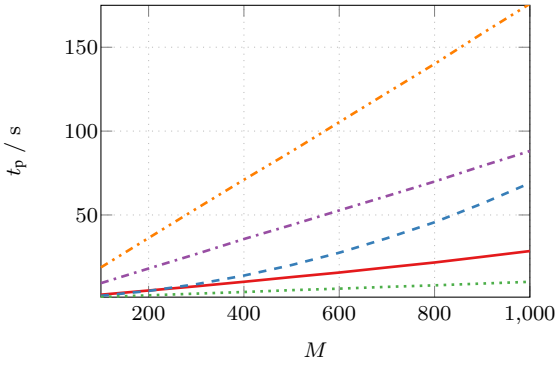

Fig. 2. Comparison of the processing time t_{p} for the proposed smoother (——), FFBSi (– – –), KSD (·····), CPF-AS with $K = 10$ trajectories (–··–··–), and CPF-AS with $K = 20$ trajectories (–·––·––·–).

introduced in Section IV is used in the forward pass of all the smoothers. In total, $L = 100$ completely randomized Monte Carlo simulations with $N = 100$ time samples are run.

The smoothers are compared in terms of the time-averaged root mean squared error (RMSE) defined as

$$e_{\text{RMSE}} = \sqrt{\frac{1}{LN}\sum_{l=1}^{L}\sum_{n=1}^{N}(\hat{x}_{n|N}^{l} - x_n^l)^{\mathsf{T}}(\hat{x}_{n|N}^{l} - x_n^l)},$$

where the superscript l denotes the lth Monte Carlo simulation, as well as the processing time t_{p} for the complete batch of data. The methods are implemented as m-code in MATLAB and the simulations are run on a 3.4 GHz 3rd generation Intel E3 processor with 16 GB RAM.

B. Results

Fig. 1 shows the time-averaged RMSE as a function of the number of particles obtained from the Monte Carlo simulations. The figure indicates that there is no significant performance difference between the proposed method, the FFBSi smoother as well as the CPF-AS (with $K = 20$ trajectories sampled from the posterior) while the CPF-AS (with $K = 10$ trajectories) and the KSD smoothers perform somewhat worse. The latter is as expected since this was already pointed out in [37]. Note that the RMSE does not decrease significantly for more than approximately $M = 500$ particles. This is mainly due to the quite smooth and weak non-linearity.

Fig. 2 shows the measured processing time for all five smoothers with varying number of particles. Since these numbers depend on the specific implementation as well as the platform, their significance is mainly qualitative rather than quantitative. Unlike for the RMSE, the differences here are more significant. The proposed smoother is roughly four times faster than the CPF-AS (for $K = 10$ trajectories) and about as fast as the FFBSi smoother for low numbers of particles and considerably faster for larger M. The KSD smoother is the fastest, being roughly two to three times as fast as the proposed method. However, thanks to the efficient implementation, the computational complexity increases much more slowly for the proposed smoother compared to, for example, the FFBSi smoother. The CPF-AS and KSD smoothers scale linearly, as expected [37], [42].

VII. Conclusions

In this paper, we proposed a particle smoother for Wiener state-space models based on the two-filter formulation. In contrast to existing two-filter smoothers, the proposed method exploits the model structure to obtain a closed-form expression for the backward dynamics. Furthermore, auxiliary particle filters where the optimal proposal densities are approximated using Gaussian densities are used in both filters.

The numerical results indicate that the proposed method performs as well as the compared state of the art smoothers in terms of RMSE. This is to be expected since all methods target the same smoothing density which is indeed approximated similarly by all the compared algorithms. The advantage of the proposed smoother over the compared ones is highlighted in the comparison of the computational complexity. The two-filter structure of the proposed smoother only requires two filters to be run (plus a few light-weight computations in propagating the prior mean and covariance as well as calculating the smoothed weights) and thus, the smoother scales well in terms of the number of particles and time samples, making it suitable in applications that are computationally prohibitive for other smoothers.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[2] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Handbook of Nonlinear Filtering*, ser. Oxford Handbooks, D. Crisan and B. Rozovskii, Eds. Oxford, UK: Oxford University Press, 2011, vol. 12, pp. 656–704.

[3] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[4] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.

[5] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[6] T. B. Schön, F. Gustafsson, and R. Karlsson, "The particle filter in practice," in *Handbook of Nonlinear Filtering*, ser. Oxford Handbooks, D. Crisan and B. Rozovskii, Eds. Oxford, UK: Oxford University Press, 2011, vol. 12.

[7] R. Hostettler and P. M. Djurić, "Vehicle tracking based on fusion of magnetometer and accelerometer sensor measurements with particle filtering," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 4917–4928, November 2015.

[8] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao–Blackwellized particle filter for multiple target tracking," *Information Fusion*, vol. 8, no. 1, pp. 2–15, 2007.

[9] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and Statistics: Essays in Honor of David A. Freedman*, ser. Collections, D. Nolan and T. Speed, Eds. Institute of Mathematical Statistics, 2008, vol. 2, pp. 316–334.

[10] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[11] R. van der Merwe, A. Doucet, N. de Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in Neural Information Processing Systems*, 2001, pp. 584–590.

[12] C. Naesseth, F. Lindsten, and T. B. Schön, "Nested sequential Monte Carlo methods," in *32nd International Conference on Machine Learning (ICML)*, Lille, France, July 2015, pp. 1292–1301.

[13] P. Bunch and S. J. Godsill, "Approximations of the optimal importance density using Gaussian particle flow importance sampling," *Journal of the American Statistical Association*, vol. 111, no. 514, pp. 748–762, 2016.

[14] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao–Blackwellised particle filtering for dynamic Bayesian networks," in *16th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2000, pp. 176–183.

[15] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, July 2005.

[16] F. Lindsten, P. Bunch, S. Särkkä, T. B. Schön, and S. J. Godsill, "Rao–Blackwellized particle smoothers for conditionally linear Gaussian models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 353–365, March 2016.

[17] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. Wiley, April 1980.

[18] Y. Bresler, "Two-filter formulae for discrete-time non-linear Bayesian smoothing," *International Journal of Control*, vol. 43, no. 2, pp. 629–641, 1986.

[19] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state–space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010.

[20] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[21] M. Roth, G. Hendeby, and F. Gustafsson, "Nonlinear Kalman filters explained: A tutorial on moment computations and sigma point methods," *Journal of Advances in Information Fusion*, vol. 11, no. 1, pp. 47–70, 2016.

[22] M. Briers, "Improved Monte Carlo methods for state-space models," Ph.D. dissertation, University of Cambridge, 2007.

[23] R. Hostettler, "A two filter particle smoother for Wiener state-space systems," in *IEEE Conference on Control Applications (CCA)*, Sydney, Australia, September 2015.

[24] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[25] P. E. Jacob, L. M. Murray, and S. Rubenthaler, "Path storage in the particle filter," *Statistics and Computing*, vol. 25, no. 2, pp. 487–496, March 2015.

[26] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, August 1965.

[27] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.

[28] R. Douc, A. Garivier, E. Moulines, and J. Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models," *The Annals of Applied Probability*, vol. 21, no. 6, pp. 2109–2145, December 2011.

[29] E. Taghavi, F. Lindsten, L. Svensson, and T. B. Schön, "Adaptive stopping for fast particle smoothing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, May 2013, pp. 6293–6297.

[30] P. Bunch and S. J. Godsill, "Improved particle approximations to the joint smoothing distribution using Markov Chain Monte Carlo," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 956–963, February 2013.

[31] F. Lindsten and T. B. Schön, "Backward simulation methods for Monte Carlo statistical inference," *Foundations and Trends in Machine Learning*, vol. 6, pp. 1–143, 2013.

[32] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, August 1969.

[33] G. Kitagawa, "The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother," *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.

[34] M. Hürzeler and H. R. Künsch, "Monte Carlo approximations for general state-space models," *Journal of Computational and Graphical Statistics*, vol. 7, no. 2, pp. 175–193, 1998.

[35] M. Briers, A. Doucet, and S. S. Singh, "Sequential auxiliary particle belief propagation," in *7th International Conference on Information Fusion (FUSION)*, vol. 1, July 2005.

[36] T. N. M. Nguyen, S. Le Corff, and E. Moulines, "On the two-filter approximations of marginal smoothing distributions in general state space," *Advances in Applied Probability*, March 2018.

[37] J. Kronander, T. B. Schön, and J. Dahlin, "Backward sequential Monte Carlo for marginal smoothing," in *IEEE Workshop on Statistical Signal Processing (SSP)*, June 2014, pp. 368–371.

[38] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[39] N. Whiteley, "Discussion on particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B*, vol. 72, pp. 306–307, 2010.

[40] N. Whiteley, C. Andrieu, and A. Doucet, "Efficient bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods," Bristol Statistics Research, Tech. Rep., 2010.

[41] F. Lindsten, M. I. Jordan, and T. B. Schön, "Particle Gibbs with ancestor sampling," *Journal of Machine Learning Research*, vol. 15, pp. 2145–2184, 2014.

[42] A. Svensson, T. B. Schön, and M. Kok, "Nonlinear state space smoothing using the conditional particle filter," in *17th IFAC Symposium on System Identification (SYSID)*, vol. 48, no. 28, Beijing, China, 2015, pp. 975–980.

[43] J. Westerborn and J. Olsson, "Efficient particle-based online smoothing in general hidden Markov models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8003–8007.

[44] J. Olsson and J. Westerborn, "Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm," *Bernoulli*, vol. 23, pp. 1951–1996, August 2017.

[45] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[46] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[47] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.

[48] J. H. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, October 2003.

[49] F. Lindsten, T. B. Schön, and M. I. Jordan, "Bayesian semiparametric Wiener system identification," *Automatica*, vol. 49, no. 7, pp. 2053–2063, 2013.