Particle-based adaptive-lag online marginal smoothing in general state-space models

Johan Alenløv and Jimmy Olsson

Abstract—We present a novel algorithm, an adaptive-lag smoother, approximating efficiently, in an online fashion, sequences of expectations under the marginal smoothing distributions in general state-space models. The algorithm evolves recursively a bank of estimators, one for each marginal, in resemblance with the so-called particle-based, rapid incremental smoother (PaRIS). Each estimator is propagated until a stopping criterion, measuring the fluctuations of the estimates, is met. The presented algorithm is furnished with theoretical results describing its asymptotic limit and memory usage.

Index Terms—Sequential Monte Carlo methods, state-space models, marginal smoothing, PaRIS, particle filters, state estimation.

I. INTRODUCTION

STATE-SPACE models (SSMs), also known as hidden Markov models (HMMs), are fundamental in many scientific and engineering disciplines. Incorporating unobservable, Markovian states, these models are adjustable enough to model a variety of complex, real-world time series in, e.g., econometrics [4], speech recognition [23], and target tracking [17]. In this paper we focus on online state reconstruction in SSMs; more specifically, our goal is to estimate, on-the-fly as new data appear, expectations under the marginal posteriors of the different states given the data.

More precisely, an SSM is a bivariate model consisting of an observable process \( \{Y_t\}_{t \in \mathbb{N}} \), referred to as the observation process, and an unobservable Markov chain \( \{X_t\}_{t \in \mathbb{N}} \), known as the state process, taking on values in some general state spaces \((Y, Y)\) and \((X, X')\), respectively. Throughout the paper the subindex \( t \) will often be referred to as “time” without being necessarily a temporal index.

When operating on SSMs one is often interested in calculating expectations under the conditional distribution of one or several states conditioned upon a subset of some given stream \( \{y_t\}_{t \in \mathbb{N}} \) of observations. For any \((s, s', t) \in \mathbb{N}^3\) such that \(s \leq s' \leq t\) we denote by \(\phi_{s:s'|t}\) the conditional distribution of \(X_{s:s'} = (X_s, \ldots, X_{s'})\) (our notation for vectors) given \(Y_{0:t} = y_{0:t}\) (a precise definition is given in Section II).

Some special distributions of interest are the filter distributions \(\phi_t := \phi_{t:t}|t\), the joint-smoothing distributions \(\phi_{s:t}|t\), and, finally, when \(s = s'\), the marginal smoothing distributions \(\phi_{s|t}\).

For any probability measure \(\mu\) and real-valued measurable function \(h\), let \(\mu h := \int h(x) \mu(dx)\) denote the Lebesgue integral of \(h\) with respect to \(\mu\) (whenever this is well defined). Given an observation stream \(\{y_t\}_{t \in \mathbb{N}}\) and a sequence \(\{h_t\}_{t \in \mathbb{N}}\) of real-valued functions, the present paper focuses on online calculation of the flow

\[
(\phi_0|t h_0, \phi_1|t h_1, \ldots, \phi_{s-1}|t h_{s-1}, \phi_t|t h_t), \quad t \in \mathbb{N},
\]

as \(t\) increases. Note that a significant challenge with this problem is that all elements of the vector \((\phi)\) change with \(t\). Moreover, in the problem formulation, the word “online” means, first, that the observation sequence should be processed in a single sweep as new observations appear and, second, that the computational cost and memory requirements of the algorithm should stay constant over time.

A. Previous work

Exact computations of the marginal smoothing distributions is possible only in models with finite state space and for linear Gaussian SSMs using the forward-backward smoother [22] and the disturbance smoother [6] (sometimes referred to as the Kalman smoother), respectively. When dealing with general, possibly nonlinear SSMs, current methods take mainly two different approaches. The first approach relies, as in the extended Kalman filter and the uncentred Kalman filter [11], [15], on linearisation of the model and Kalman filtering. These methods work well if the model is almost linear Gaussian, but will introduce significant errors in the presence of highly nonlinear model components.

The second approach relies on Monte Carlo simulation, preferably in the form of sequential Monte Carlo (SMC) methods, or, particle filters. Particle filters propagate recursively a sample of random simulations, so-called particles, mimicking typically, as in the bootstrap particle filter, the latent state (or, prior) dynamics. At each time step, the particles are associated with importance weights compensating for the discrepancy between the prior dynamics and the conditional (posterior) dynamics of the states given the observations. By duplicating and killing, through resampling, particles with high and low importance, respectively, the particles are directed towards regions of the state space with high posterior probability.

At each time step, the weighted empirical measure associated with the particles serves as an approximation of the filter distribution at the time step in question. Moreover, it can be shown that the empirical measure formed by the ancestral lines of the particles provides an approximation also of the joint smoothing distribution. However, since the particles are resampled repeatedly, these trajectories collapse eventually, and in the long run such a Monte Carlo approximation will
rely on more or less a single path. Thus, the statistician is referred to alternative techniques such as the forward-filtering backward-smoothing (FFBSm) \[11\] and the forward-filtering backward-simulation (FFBSi) algorithms \[12\], which approximate the so-called backward decomposition of the joint-smoothing distribution, or Markov chain Monte Carlo (MCMC) methods simulating a Markov chain admitting the joint-smoothing distribution as invariant distribution. Recently, \[5\] combined successfully particle methods and MCMC into particle Markov chain Monte Carlo (PMCMC) methods to construct such Markov chains. In their most basic form, the FFBSi and FFBSm algorithms are however two-pass algorithms in the sense that they first approximate, forwards in time, the filter distributions, whereupon smoothing is performed in the backward direction. Thus, the online processing setting of the present paper invalidates these methods. The same can be said about the MCMC and PMCMC methods, which design Markov chains targeting the joint-smoothing distribution for a fixed \(t\).

As known to us, there is only one existing approach that does not violate the online criteria set up by us, namely the fixed-lag smoother \[16\], \[19\], which is a genealogical tracing-based particle method that copes with particle path degeneracy by means of truncation. The truncation implies a bias that may be controlled using forgetting arguments (see \[19\] for an analysis). A problem with this method, which will be explained in some detail in Section IV-B, is that the truncation lag is a design parameter that needs to be set a priori by the user. This is nontrivial, as the optimal choice of the lag depends on the mixing properties of the model.

B. Our Contribution

In the present paper we introduce a novel algorithm that provides an approximate solution to the research question outlined in the previous section. The algorithm relies on a recursive form of the backward decomposition that has been employed previously for forward smoothing of additive state functionals \[2\], \[9\]. Of special interest is the particle-based, rapid incremental smoother (PaRIS), presented by the authors in \[20\] and analysed theoretically in \[21\], which performs online smoothing of additive state functionals with constant memory and computational complexity demands that grow only linearly with the number of particles, compared with the quadratic growth of other algorithms. The aim of this paper is to extend the PaRISian methodology to the problem of estimating expectation flows of the form \(g\), leading to an adaptive-lag approach.

Our contribution is presented in two steps: first, we consider an ideal algorithm, applicable in the context of linear Gaussian models and requiring closed-form computation of all quantities of interest; after this, the general—possible nonlinear/non-Gaussian—case is dealt with using particle-based approximations. By using results derived in \[21\], we are able to analyse theoretically the asymptotic properties and memory demands of our algorithm in order to place the same on more solid ground.

C. Outline

In Section II we introduce formally SSMs and the backward decomposition. Section III introduces the ideal algorithm, providing a conceptual understanding of our approach without involving particle approximations. In Section IV we turn to the nonlinear/non-Gaussian case and replace intractable quantities by particle-based estimates. Theoretical results are presented in Section V and in Section VI the algorithm is benchmarked numerically on two models. Our conclusions are presented in Section VII and, finally, Section A contains some proofs.

II. Preliminaries

In the following, let \(\mathbb{N}\) and \(\mathbb{N}^{+}\) denote the sets of nonnegative and positive natural numbers, respectively. For any bounded measurable function \(h\) we let \(\|h\|_{\infty} := \sup_{x \in \mathcal{X}} |h(x)|\) and \(\text{osc}(h) := \sup_{(x, x') \in \mathcal{X}^2} |h(x) - h(x')|\) denote the supremum and oscillator norms of \(h\), respectively.

In the following we assume that all random elements are well-defined on a common probability space \((\Omega, \mathcal{F}, \mathbb{P})\). Let \((\mathcal{X}, \mathcal{X})\) and \((\mathcal{Y}, \mathcal{Y})\) be some measurable spaces, \(\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \to [0, 1]\) and \(\mathbf{G} : \mathcal{X} \times \mathcal{Y} \to [0, 1]\) some Markov transition kernels, and \(\chi\) a probability measure on \(\mathcal{X}\). An SSM is a bivariate Markov chain \(((X_t, Y_t))_{t \in \mathbb{N}}\) on \(\mathcal{X} \times \mathcal{Y}\) such that \(X_{t+1}, Y_{t+1} \mid X_t, Y_t \sim \mathbf{Q}(X_t, dx_{t+1}) \mathbf{G}(x_{t+1}, dy_{t+1})\) and \(X_0, Y_0 \sim \chi(dx_0) \mathbf{G}(x_0, dy_0)\). It is assumed that only \(\{Y_t\}_{t \in \mathbb{N}}\) is observable. Using this definition, it is easily shown that

(i) the unobservable state sequence \(\{X_t\}_{t \in \mathbb{N}}\) is a Markov chain with transition kernel \(\mathbf{Q}\) and initial distribution \(\chi\).

(ii) the variables of the observable process \(\{Y_t\}_{t \in \mathbb{N}}\) are, conditionally on the states, independent and such that the conditional distribution of each \(Y_t\) depends on the corresponding \(X_t\) only.

Throughout this paper we will assume that the model is fully dominated, i.e., that \(\mathbf{Q}\) and \(\mathbf{G}\) admit transition densities \(q\) and \(g\), respectively, with respect to some reference measures. We will in the following assume that we are given a fixed stream \(\{y_t\}_{t \in \mathbb{N}}\) of observations. For ease of notation, let for all \(t \in \mathbb{N}\), \(g_t(x) := g(x, y_t), x \in \mathcal{X}\). The joint smoothing distribution at time \(t\), i.e., the law of \(X_{0:t}\) conditionally to \(Y_{0:t} = y_{0:t}\), is

\[
\phi_{0:t}(dx_{0:t}) := \frac{\chi(dx_0) q_0(x_0) \prod_{s=1}^t q(x_{s-1}, dx_s) g_s(x_s)}{\int \chi(dx_0) q_0(x_0) \prod_{s=1}^t q(x_{s-1}, dx_s) g_s(x_s)}
\]

and all posteriors \(\phi_{s:t}^{x,y}\) of interest (including the filter \(\phi_t\)) are obtained as marginals of \(\phi_{0:t}^{x,y}\).

Interestingly, the state process is still Markov when evolving conditionally to \(Y_{0:t}\) in the time-reversed direction; in particular, the distribution of \(X_s\) given \(X_{s+1} = x_{s+1}\) and \(Y_{0:s} = y_{0:s}\) is given by

\[
\tilde{Q}_{\phi_s}(x_{s+1}, dx_s) := \frac{q(x_{s+1}, x_{s+1}) \phi_s(dx_s)}{\int q(x_{s+1}, x_{s+1}) \phi_s(dx_s)}, \quad s \in \mathbb{N}.
\]

(2) (see, e.g., \[3\] Prop. 3.3.6). Using (2), the joint-smoothing distribution may be expressed by the backward decomposition

\[
\phi_{0:t}(dx_{0:t}) = \phi_t(dx_t) \prod_{s=0}^{t-1} \tilde{Q}_{\phi_s}(x_{s+1}, dx_s),
\]

which is instrumental in many smoothing procedures \[12\], \[10\], \[21\], \[9\].
### III. ONLINE MARGINAL SMOOTHING

Recall that our aim is to estimate the vectors \( \{ \phi_t \} \) in an online manner as \( t \) increases. For the moment, consider estimation of some expectation \( \phi_s \| h_s \) under some marginal \( \phi_s \). Under suitable ergodicity conditions (to be specified later), we may expect observations of the distant future to have limited effect on the posterior of some state \( X_s \). Consequently, we may expect \( \phi_s \| h_s \) to converge to some fixed point \( \phi_s \) as \( t \) increases; see [3] Sec. 4.3. Thus, allowing for a negligible bias, we may update \( \phi_s \| h_s \) only as long as the sequence \( \{ \phi_s \| h_s \} \) exhibits discernible fluctuations, i.e., until, say, \( t = s_\varepsilon \), and approximate \( \phi_s \| h_s \) by \( \phi_s \) for all \( t \geq s_\varepsilon \). Here \( \varepsilon \) is an algorithmic parameter regulating the stopping criterion. This idea is explored in the following.

#### A. An ideal algorithm

Let \(( s, t ) \in \mathbb{N} \) be such that \( s \leq t \) and consider the marginal expectation \( \phi_s \| h_s \) as \( \mathbb{E}[h_s(X_s) \mid Y_{0:t} = y_{0:t}] \). By the tower property,

\[
\mathbb{E}[h_s(X_s) \mid Y_{0:t} = y_{0:t}] = \mathbb{E}[T_{s|t}(X_t) \mid Y_{0:t} = y_{0:t}],
\]

where

\[
T_{s|t}(x_t) := \mathbb{E}[h_s(X_s) \mid Y_{0:t-1} = y_{0:t-1}, X_t = x_t],
\]

\( x_t \in X \),

is a statistic appearing frequently in the literature on smoothing; see for instance [21], [10], [9]. Appealingly, reapplying the tower property, the statistics \( \{ T_{s|t} \}_{t \geq s} \) can be expressed recursively (see [13], [2], [8]) through

\[
T_{s|t+1}(x_{t+1}) = \mathbb{E}[T_{s|t}(X_t) \mid Y_{0:t} = y_{0:t}, X_{t+1} = x_{t+1}]
= \int T_{s|t}(x_t) \phi_t(x_{t+1} \mid d x_t)
= \int T_{s|t}(x_t) \frac{q_t(x_t, x_{t+1}) \phi_t(d x_t)}{q_t(x_t, x_{t+1}) \phi_t(d x_t)}, \quad x_{t+1} \in X.
\]

The recursion is initialised by setting

\[
T_{s|s}(x_s) := h_s(x_s), \quad x_s \in X,
\]

and for completeness we define

\[
T_{s|u} = 0 \text{ for } u < s.
\]

By (3), \( \phi_s \| h_s = \phi_t T_{s|t} \), and the target can hence be calculated by applying the filter \( \phi_t \) to the function \( T_{s|t} \).

Now the question arises when to stop updating the quantity of interest; indeed, since we are interested in computing the full vector \( \{ \phi_t \} \) but opposed to letting the computational complexity of the algorithm increase with time, we are forced to terminate updating when the fluctuations of the sequence \( \{ \phi_t T_{s|t} \} \) have ceased. In the present paper we will stop updating at the time point \( s_\varepsilon \) for which the variance of \( T_{s|t} \) under the filter \( \phi_t \) falls below some given threshold \( \varepsilon > 0 \) for the first time. After that, we output \( \phi_{s\varepsilon} T_{s\varepsilon} \), as our estimate of \( \phi_t T_{s|t} \) for all \( t \geq s_\varepsilon \). This choice can be clearly motivated by (5), from which it is clear that once \( T_{s|t} \) is close to constant in the support of \( \phi_t \), then also \( T_{s|t+1} \) is close to constant everywhere.

The algorithm can be summarised as follows.

- Initialise by letting \( S \leftarrow \emptyset \). The set \( S \) will keep track of our active estimators. In addition, set the tolerance \( \varepsilon \).
- For \( t \leftarrow 0, 1, 2, \ldots \)
  - for each \( s \in S \), calculate \( T_{s|t}(x_t) \) using (5);
  - let \( S \leftarrow S \cup \{ t \} \), i.e. activate an estimator at time \( t \) and set \( T_{s|t} \leftarrow h_t \);
  - for each \( s \in S \), calculate the variance \( \mathbb{V}[\phi_t T_{s|t}(X_t)] \); if it is smaller than \( \varepsilon \), let \( S \leftarrow S \setminus \{ s \} \) and output \( \phi_t T_{s|t} \).

#### B. Example: linear Gaussian SSMs

As mentioned previously, exact computation of the filter and joint-smoothing distributions is possible only for a few specific models. Here we present a Kalman-based version in the case of linear Gaussian SSMs.

In the linear Gaussian SSMs, an \( n_x \)-dimensional autoregressive state process is partially observed through \( n_y \)-dimensional observations. The model is specified by the equations

\[
X_{t+1} = AX_t + U_{t+1},
\]

\[
Y_t = BX_t + V_t,
\]

where \( A \in \mathbb{R}^{n_x \times n_x} \) and \( B \in \mathbb{R}^{n_y \times n_x} \) and \( \{ U_t \}_{t \in \mathbb{N}} \) and \( \{ V_t \}_{t \in \mathbb{N}} \) are sequences of mutually independent Gaussian noises with zero mean and covariance matrices \( \Sigma_U \in \mathbb{R}^{n_x \times n_x} \) and \( \Sigma_V \in \mathbb{R}^{n_y \times n_y} \), respectively. All matrices are assumed to be pre-specified. Given the sequence \( \{ y_t \}_{t \in \mathbb{N}} \) of observations we wish to estimate \( \{ \phi_t \} \) in the case of affine objective functions \( h_s(x) = \alpha_x^T x + \beta_s \), where \( \alpha_x \in \mathbb{R}^{n_x} \) and \( \beta_s \in \mathbb{R} \) are pre-specified.

In this model each filter \( \phi_t \) is Gaussian, and the Kalman filter propagates its mean \( \mu_t \) and covariance matrix \( \Sigma_t \) recursively through time. We calculate the backward kernel \( (2) \), which is proportional to the filter at time \( s \) times the transition density of the latent Markov chain. Since both these distributions are Gaussian with known mean and covariance matrices, it is easy to show that also the distribution of \( X_t \) conditioned on \( X_{t+1} = x_{t+1} \) and \( Y_{0:t} = y_{0:t} \) is Gaussian with mean \( \mu_{t|t+1}(x_{t+1}) = \Sigma_{t|t+1} (A^T \Sigma_{t+1}^{-1} x_{t+1} + \Sigma_t^{-1} \mu_t) \) and covariance matrix \( \Sigma_{t|t+1} = (A^T \Sigma_{t+1}^{-1} A + \Sigma_t^{-1})^{-1} \). We may hence write down a specific updating procedure for the functions \( \{ T_{s|t} \}_{t \geq s} \) in this case.

- Initialisation: for \( t = s \), let \( T_{s|s}(x_s) = h_s(x_s) = \alpha_s^T x_s + \beta_s \).
- Proceeding recursively, assume that \( \alpha_s \) and \( \beta_s \) are of form \( T_{s|t}(x_t) = \alpha_s^T x_t + \beta_s \); then \( T_{s|t+1}(x_{t+1}) = \alpha_{s|t+1}^T x_{t+1} + \beta_{s|t+1} \), where
  \[
  \alpha_{s|t+1} = \alpha_s^T \Sigma_{t+1}^{-1} A + \Sigma_t^{-1} \mu_t + \beta_s,
  \]
  \[
  \beta_{s|t+1} = \alpha_s^T \Sigma_{t+1}^{-1} A + \Sigma_t^{-1} \mu_t + \beta_s.
  \]

The last step of the algorithm—consisting in checking whether the variance of the function above is small enough—is now easily performed by calculating \( \mathbb{V}[\phi_t T_{s|t}(X_t)] = \)}
for all \( \alpha^T \Sigma_i \alpha_{i,i} \) and comparing this with some pre-specified threshold \( \varepsilon \). This completes all the steps needed for executing the algorithm, and we refer to Section VI for a numerical illustration.

Even though the previous provides an exact implementation of the algorithm, it is limited to a single class of models and specific target functions. To move beyond this simplified setting we need to rely on approximations, and this will be discussed in the next section.

IV. PARTICLE-BASED ONLINE MARGINAL SMOOTHING

As mentioned previously, exact computation of the filter distributions—and hence the backward kernels—is possible only in a few specific cases. In the general case we will approximate these distributions using particle filters, which are recalled in the following.

A. Particle filters

A particle filter propagates recursively a set of particles with associated weights in order to approximate the filter distribution flow \( \{ \phi_t \}_{t \in \mathbb{N}} \) given the sequence \( \{ y_t \}_{t \in \mathbb{N}} \).

We describe recursively the most basic particle filter—the bootstrap filter—and assume that we have at hand a sample \( \{ (\xi_{i,t}, \omega_i) \}_{i=1}^{N} \) of particles (the \( \xi_{i,t} \)) and associated weights (the \( \omega_i \)) targeting the filter distribution \( \phi_t \) in the sense that for all \( \phi_t \)-integrable functions \( f \),

\[
\sum_{i=1}^{N} \frac{\omega_i}{\Omega_t} f(\xi_{i,t}) \approx \phi_t f, \quad \text{as } N \to \infty,
\]

where \( \Omega_t := \sum_{i=1}^{N} \omega_i \) denotes the total weight. To form a new weighted sample \( \{ (\omega_{i,t+1}, \xi_{i,t+1}) \}_{i=1}^{N} \) targeting the subsequent filter distribution \( \phi_{t+1} \), a two-step procedure is applied. First, the particles are resampled by drawing randomly a set of \( N \) independent indices \( \{ I_{t+1}^{i} \}_{i=1}^{N} \) from the categorical distribution induced by the probabilities proportional to the weights \( \{ \omega_{i,t+1} \}_{i=1}^{N} \), an operation denoted by \( I_{t+1}^{i} \sim \text{Pr}(\{ \omega_{i} \}_{i=1}^{N}) \), \( i \in \{1, \ldots, N\} \).

Second, the resampled particles are moved conditionally independently according to the dynamics of the state process, i.e., for all \( i \in \{1, \ldots, N\} \),

\[
\xi_{i,t+1} \sim Q(\xi_{i,t+1}^{I_{t+1}}).
\]

Finally, new importance weights are computed according to

\[
\omega_{i,t+1} = g_{t+1}(\xi_{i,t+1})
\]

for all \( i \in \{1, \ldots, N\} \).

Initialisation is carried through by drawing \( \xi_{i,0} \sim \chi \) and setting \( \omega_{i} = g_0(\xi_{i,0}) \) for all \( i \in \{1, \ldots, N\} \).

We summarise the procedure in Algorithm 1 and denote by \( \text{PF}(\{ (\xi_{i,t}, \omega_i) \}_{i=1}^{N}, y_{t+1}) \) one application of Algorithm 1 (By convention we let \( \{ (\xi_{0}, \omega_0) \}_{i=1}^{N} \sim \text{PF}(\{ (\xi_{1-1}, \omega_{1-1}) \}_{i=1}^{N}, y_0) \) denote the initial step.

So far we have only considered estimation of the filter distributions. The rest of this section will be devoted to particle approximation of the marginal smoothing distributions.

Algorithm 1 Bootstrap particle filter

Require: A weighted sample \( \{ (\xi_{i,t}, \omega_i) \}_{i=1}^{N} \) targeting the filter distribution \( \phi_t \)
1: for \( i = 1 \to N \) do
2: \( I_{t+1}^{i} \sim \text{Pr}(\{ \omega_{i} \}_{i=1}^{N}); \)
3: \( \xi_{i,t+1} \sim Q(\xi_{i,t+1}^{I_{t+1}}); \)
4: \( \omega_{i,t+1} = g_{t+1}(\xi_{i,t+1}) \)
5: end for
6: return \( \{ (\xi_{i,t+1}, \omega_i^{i,t+1}) \}_{i=1}^{N} \)

For all \( i \in \{1, \ldots, N\} \), define recursively genealogical indices \( \{ G_{s|t}^{i} \}_{s=0}^{t} \) by \( G_{t|t}^{i} = i \) and \( G_{s-1|t}^{i} = I_{s}^{G_{s|t}^{i}} \). The set \( \{ (\xi_{0}, \ldots, \xi_{t}) \}_{i=1}^{N} \) is often referred to as the genealogical tree of the particles, and it is easy to show that the genealogical tree may, together with the importance weights \( \{ \omega_i \}_{i=1}^{N} \), be used for estimating the joint-smoothing distribution. In particular,

\[
\sum_{i=1}^{N} \frac{\omega_i}{\Omega_t} h_s(\xi_{t}^{G_{s|t}^{i}}) \approx \phi_{s|t} h_s, \quad \text{as } N \to \infty. \tag{6}
\]

We refer to this estimator as the Poor man’s smoother. A well-known problem with the Poor man’s smoother is that the repeated resampling operations of the particle filter always deplete the genealogical tree in the long run; thus, sooner or later, for some \( s < t \) and some \( i_0 \), \( G_{u|t}^{i_0} = G_{u|t}^{j} \) for all \( j \in \{1, \ldots, N\} \) and all \( u \in \{0, \ldots, s\} \), implying that the estimates of \( \phi_{u|t} h_s, u \in \{0, \ldots, s\} \), will be based on only a single particle path. The work [13] establishes, under assumptions requiring typically the state space to be a compact set, a bound on the expected distance from the last generation to the most recent common ancestor that is proportional to \( N \log(N) \) and uniform in time. Thus, the number of active, unique particles in the estimator (6) tends to one as \( t \) increases, leading to a depleted and impractical estimator.

B. Fixed-lag smoothing

To remedy the problem of particle lineage degeneracy, a fixed-lag smoother [16] can be used. The idea is to approximate the marginal smoothing distribution \( \phi_{s|t} \) by the distribution \( \phi_{s|\Delta(t)} \) where \( s_{\Delta}(t) = (s + \Delta) \wedge t \) for some pre-specified lag \( \Delta \in \mathbb{N}^{*} \). In this case, using the notation above, we get the biased approximation

\[
\phi_{s|t} h_s \approx \sum_{i=1}^{N} \frac{\omega_i}{\Omega_s} h_s(\xi_{s_{\Delta}(t)}^{G_{s_{\Delta}(t)}^{i}}),
\]

where \( G_{s_{\Delta}(t)}^{i} \) is defined as above.

The approach requires suitable design of the lag \( \Delta \), and we face here a classical bias-variance tradeoff: if \( \Delta \) is too small, then the forgetting of the model has not kicked in, and the discrepancy between the distributions is going to be large (leading to high bias); on the other hand, if \( \Delta \) is too large, then, by path degeneracy, the estimate will be depleted (leading to high variance). The optimal choice of lag depends on the mixing of the SSM, and [19] proposes an optimal choice of
Δ as \([c \log(t)]\), where the constant \(c\) depends on the mixing. This is problematic since it is hard to calculate, and even estimate, the mixing of an SSM. Thus, designing properly the lag is indeed a non-trivial task. The lag-based particle estimator that we propose in the next section relies again on forgetting-based arguments, but adapts the lag in a completely automatic manner.

C. The adaptive-lag smoother

We present here a particle-based version of the ideal algorithm in Section III-A which can be thought of as an adaptive-lag smoother. In this algorithm we employ novel techniques for updating particle estimates of the functions \(T_s[t]\), and the adaptive lag can be thought of as the number of steps that each function is updated before the stopping criterion triggers truncation. The truncation depends heavily on the mixing of the model, but is now determined in an adaptive manner.

The crucial in the algorithm presented in Section III-A is the need of estimating and updating the statistics \(T_s[t]\) through the recursion (5). We proceed by induction and assume that we have at hand a set \(\{\hat{\tau}_s[t]\}_{i=1}^{N}\) of estimates of \(\{T_s[t](\xi_t^i)\}_{i=1}^{N}\). Proceeding as in [9], these estimates are updated to estimates \(\{\hat{\tau}_s[t+1]\}_{i=1}^{N}\) of \(\{T_s[t+1](\xi_t^{i+1})\}_{i=1}^{N}\) by replacing, in (5), \(\phi_t\) by a particle approximation, yielding

\[
\hat{\tau}_s[t+1] = \frac{N}{s=1} \sum_{s=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{s=1}^{N} \omega_t^i q(\xi_t^i, \xi_t^{i+1})} \hat{\tau}_s[t] \quad i \in \{1, \ldots, N\},
\]

where the ratio is a particle approximation of the backward kernel (2), and the particle approximation \(\sum_{s=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{s=1}^{N} \omega_t^i q(\xi_t^i, \xi_t^{i+1})}\) is linear in \(\phi_t\). Casting the recursion (7) into the ideal algorithm in Section III yields the following procedure:

1. Initialise by letting \(S \leftarrow \emptyset\) and setting the tolerance \(\varepsilon\).
2. For \(t \leftarrow 0, 1, 2, 3, \ldots\)
   - run \(\{(\xi_t^i, \omega_t^i)\}_{i=1}^{N} \leftarrow \text{PF}(\{(\xi_{t-1}^i, \omega_{t-1}^i)\}_{i=1}^{N}, y_t)\);
   - for each \(s \in S\) and \(i \in \{1, \ldots, N\}\), calculate \(\hat{\tau}_s[t]\) using (7);
   - let \(S \leftarrow S \cup \{t\}\) and \(\hat{\tau}_s[t] \leftarrow h_t(\xi_t^i)\) for all \(i \in \{1, \ldots, N\}\).
3. For each \(s \in S\), if
   \[
   \sum_{i=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{s=1}^{N} \omega_t^i q(\xi_t^i, \xi_t^{i+1})} \hat{\tau}_s[t] < \varepsilon,
   \]
   then let \(S \leftarrow S \setminus \{s\}\) and output \(\sum_{i=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{s=1}^{N} \omega_t^i q(\xi_t^i, \xi_t^{i+1})} \hat{\tau}_s[t] / \Omega_t\) as an estimate of \(\phi_{t|t} h_s\) for all \(t' \geq t\).

A drawback with the updating formula (7) is that it requires a sum of \(N\) terms to be computed for each particle, which yields an overall \(O(N^2)\) computational complexity. Needless to say, this is impractical when \(N\) is large.

To reduce the computational burden, we proceed as in the PaRIS [21, Alg. 2] and replace the right hand side of (7), which can be interpreted as an expectation, by a Monte Carlo estimate. More precisely, assuming that we have at hand a set \(\{\tau_s[t]\}_{i=1}^{N}\) of estimates of \(\{T_s[t](\xi_t^i)\}_{i=1}^{N}\), we replace (7) by the mean

\[
\tau_s[t+1] = \frac{1}{N} \sum_{j=1}^{N} \tau_s[(i, j)]
\]

where \(\{\tau_s[t]\}_{j=1}^{N}\) are conditionally independent draws from \(\text{Pr}(\{\omega_t^j q(\xi_t^j, \xi_t^{j+1})\}_{j=1}^{N})\) and \(\tilde{N}\) is a precision parameter. Such draws can most often be produced at low computational cost using rejection sampling. Indeed, assume that the transition density \(q(x, x')\) is uniformly bounded by some constant \(\varepsilon\), i.e., \(q(x, x') \leq \varepsilon\) for all \((x, x') \in \mathbb{X}^2\); then, following [10], a draw \(J_s[t]\) from \(\text{Pr}(\{\omega_t^j q(\xi_t^j, \xi_t^{j+1})\}_{j=1}^{N})\) can be produced by repeating the following steps until acceptance:

1. Draw \(J_s[t] \sim \text{Pr}(\{\omega_t^j\}_{j=1}^{N})\);
2. Accept \(J_s[t]\) with probability \(q(\xi_t^j, \xi_t^{j+1})/\varepsilon\).

Interestingly, under the mixing assumptions given in Section V it is possible to show that the expected number of trials required for sampling all the indices \(J_s[t]\) is in \(\tilde{N}\); see [21, Thm. 10], which yields an \(O(NN)\) algorithm, and as we will see below, \(\tilde{N}\) can be kept at a very low value (say, \(\tilde{N} = 2\)).

The variance of \(T_s[t]\) under \(\phi_t\) is estimated using

\[
\sigma_{s|t}^{2, N} := \sum_{i=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{j=1}^{N} \omega_t^j q(\xi_t^j, \xi_t^{j+1})} \tau_s[t] - \sum_{i=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{j=1}^{N} \omega_t^j q(\xi_t^j, \xi_t^{j+1})} \tau_s[t]_i^2,
\]

and the updating procedure is stopped if \(\sigma_{s|t}^{2, N} < \varepsilon\), where \(\varepsilon\) is some pre-chosen tolerance. Letting \(s_{c,t}^N(t) := \min\{s \geq 1 : \sigma_{s|t}^{2, N} < \varepsilon\} \wedge t\), we return the estimator

\[
\phi_{s_{c,t}^N} h_s := \sum_{i=1}^{N} \frac{\omega_t^i q(\xi_t^i, \xi_t^{i+1})}{\sum_{j=1}^{N} \omega_t^j q(\xi_t^j, \xi_t^{j+1})} \tau_s[t|s_{c,t}^N(t)]
\]

of \(\phi_{s_{c,t}^N} h_s\). The algorithm is presented in detail in Algorithm 2.

Algorithm 2 Adaptive-lag smoother

1. Set \(S \leftarrow \{0\}\);
2. Run \(\{(\xi_0^i, \omega_0^i)\}_{i=1}^{N} \leftarrow \text{PF}(\{(\xi_{-1}^i, \omega_{-1}^i)\}_{i=1}^{N}, y_0)\);
3. For \(i = 1 \rightarrow N\) do
   4. Set \(\tau_{0|0}^i \leftarrow h_0(\xi_0^i)\);
5. End for
6. For \(t = 1, 2, 3, \ldots\) do
   7. Run \(\{(\xi_t^i, \omega_t^i)\}_{i=1}^{N} \leftarrow \text{PF}(\{(\xi_{t-1}^i, \omega_{t-1}^i)\}_{i=1}^{N}, y_t)\);
   8. For \(i = 1 \rightarrow N\) do
      9. For \(j = 1 \rightarrow \tilde{N}\) do
         10. Draw \(J_s[t] \sim \text{Pr}(\{\omega_t^j q(\xi_t^j, \xi_t^{j+1})\}_{j=1}^{N})\);
      11. End for
   12. Set \(\tau_s[t] \leftarrow \tilde{N} - 1 \sum_{j=1}^{\tilde{N}} J_s[t|j]\);
   13. End for
   14. Set \(\hat{\tau}_s[t] \leftarrow h_t(\xi_t^i)\);
15. End for
16. Set \(S \leftarrow S \cup \{t\}\);
17. For \(s \in S\) do
18. If \(\sigma_{s|t}^{2, N} < \varepsilon\) then
19. Set \(\phi_{s|t}^{N, \varepsilon} h_s \leftarrow \sum_{i=1}^{N} \omega_t^i \hat{\tau}_s[t] / \Omega_t\) for all \(t' \geq t\);
20. Set \(S \leftarrow S \setminus \{s\}\);
21. End if
22. End for
23. End for
24. End for
D. Designing algorithmic parameters

In Algorithm 2 the parameters $\varepsilon$ and $\tilde{N}$ are set by the user. Interestingly, [21] establishes that the ParIS is (i) consistent for all fixed $\tilde{N} \in \mathbb{N}^*$ and (ii) numerically stable only if $\tilde{N} \geq 2$. The latter is illustrated by Figure 1 from which it is clear that using $\tilde{N} = 1$ leads to a degeneracy phenomenon that is reminiscent of the degeneracy of the genealogical tree. As it is clear from the same figure, this phenomenon is avoided in the case $\tilde{N} = 2$. This distinction between the cases $\tilde{N} = 1$ and $\tilde{N} = 2$ is also present in the central limit theorem in [21] Thm. 8, where, in the marginal smoothing case, a time uniform $O(1 + 1/\tilde{N})$ bound on the asymptotic variance is obtainable only in the case $\tilde{N} \geq 2$. As suggested by this bound, there is no gain in using a too large precision $\tilde{N}$, and typically $\tilde{N} = 2$ provides a satisfactory accuracy in simulations.

When it concerns the tolerance $\varepsilon$, using a smaller tolerance implies a larger lag, implying in turn more accurate estimates (and conversely). However, a larger lag requires a larger bank of active estimators, increasing in turn the computational time and memory requirement. This trade-off is studied in more detail in Section V and Section VI.

V. THEORY

A. Convergence of the sample variance criterion

We start off the theoretical analysis by studying the asymptotics (as $N \rightarrow \infty$) of the sample variances $\sigma^2_{s|t}$. The analysis will be carried through under the following assumptions.

Assumption 1. For all $t \in \mathbb{N}$, $\|g_t\|_{\infty} < \infty$. Moreover, there exists a constant $|h|_{\infty} < \infty$ such that $\text{osc}(h_t) < |h|_{\infty}$ for all $t \in \mathbb{N}$.

Assumptions similar to Assumption 2 appear frequently in the literature (see, e.g., [7]) and require typically the state space of the hidden chain to be a compact set.

Theorem 2. Under Assumptions 1 and 2, it holds, for all $(s, t) \in \mathbb{N}^2$ such that $s \leq t$ and all $\tilde{N} \in \mathbb{N}^*$,

$$\sigma^2_{s|t} \leq |h|_{\infty}^2 \left\{ c_1 \varphi^2(t-s) + c_2 \tilde{N}^{-t-s} \quad \text{if } \tilde{N} g^2 \neq 1, \\ g^2(t-s) + c_3 (t-s) \tilde{N}^{-t-s} \quad \text{if } \tilde{N} g^2 = 1, \right.$$  

(9)

where the constants $c_1$ and $c_2$ are independent of $s$ and $t$.

The first term in the bound (9) is related to the mixing of the SSM, and since $g \in (0, 1)$ this term tends to zero geometrically fast as $t$ grows. The second term is related to the Monte Carlo error induced by the ParISian update. Here we clearly see that it is required that $N \geq 2$ in order for this term to vanish as $t$ increases. In that case, $\sigma^2_{s|t} \rightarrow 0$ as $t \rightarrow \infty$.

B. Convergence of the estimator

In order to derive the asymptotic limit of our estimator we introduce the following notation. Let $s_{\varepsilon}(t) := \min\{u \geq s \geq \sigma^2_{s|u} \leq \varepsilon \wedge t$, which can be understood as the limit of $s_{\varepsilon}^N(t)$ (defined in (8)). In addition, let $\phi^t_{s|t} := \phi(s_{\varepsilon}s_{\varepsilon}(t))$ and notice that this measure differs slightly from the adaptive-lag approximation delivered by the ideal algorithm in Section III-A since the limiting variance $\sigma^2_{s|t}$ is not equal to $\forall_{\phi}(T_{s|t}(X_t))$; recall that the former also has an additional term $\eta_{s,t}$ corresponding to the Monte Carlo approximation of the backward kernel.

As expected and as established by the following theorem, $\phi^t_{s|t}$ is indeed the asymptotic limit of the proposed estimator.

Theorem 3. Let Assumptions 1 and 2 hold. Then for all $(s, t) \in \mathbb{N}^2$ such that $s \leq t$ and all bounded measurable functions $h_s$, as $N \rightarrow \infty$,

$$\phi^t_{s|t} h_s \overset{p}{\rightarrow} \phi^t_{s|t} h_s.$$  

C. Bound on asymptotic memory requirement

Finally, we show that the memory requirement of the algorithm stays, in the asymptotic regime, uniformly bounded in $t$, which was a requirement in the problem statement. Asymptotically, the estimate of $\phi^t_{s|t} h_s$ is still under construction at time $t$ if $\sigma^2_{s|t} \geq \varepsilon$ for all $t \in \{s, \ldots, t\}$. Thus, let

$$\mathcal{A}_t := \sum_{s=0}^t \prod_{u=s}^t \{ \sigma^2_{s|u} \geq \varepsilon \}$$

be the number of active adaptive-lag estimators at time $t$ in the asymptotic regime.

Theorem 4. Under Assumption 2 for all $\tilde{N} \geq 2$,

$$\mathcal{A}_t \leq \log(c/\|h\|^2 \cdot d(\varphi, \tilde{N}))/\log(c\varepsilon^2 \tilde{N}^{-1})$$

where $d(\varphi, \tilde{N}) > 0$ depends on $\varphi$ and $\tilde{N}$ only.

We remark that the bound in Theorem 4 is uniform in time, implying a uniformly bounded memory requirement of the algorithm, at least in the asymptotic regime. Moreover, we note that the bound in Theorem 4 has an $O(- \log \varepsilon)$ term.
VI. Simulations

We benchmark the algorithm on two different models. First we consider a linear Gaussian SSM, which enables computation of the exact distributions using the Kalman smoother (see, e.g., [3]). The second model is the now classical stochastic volatility model proposed in [13].

A. Linear Gaussian SSM

Consider a linear Gaussian SSM given by the following set of equations:

\[
 X_{t+1} = aX_t + \sigma_U U_{t+1}, \\
 Y_t = bX_t + \sigma_V V_t,
\]

where \( \{U_t\}_{t \geq 0} \) and \( \{V_t\}_{t \geq 0} \) are independent sequences of mutually independent standard Gaussian noise variables. Initially, \( X_0 \sim \mathcal{N}(0, \sigma_U^2/(1-a^2)) \). We consider smoothed means, corresponding to the objective function \( h_s = \text{id} \) for all \( s \in \mathbb{N} \). For this model, exact computation of the smoothing distribution is possible via the disturbance smoother; see, e.g., [3, sec. 5.2.4].

We simulate a data record comprising 201 observations \( y_0:200 \) from the model parameterised by \( (a, b, \sigma_U, \sigma_V) = (0.95, 0.5, 5, 2) \). Using tolerances \( \varepsilon \in \{1, 0.5, 0.2, 0.1\} \) and \( (N, \bar{N}) = (400, 2) \), we performed 100 independent runs of the algorithm on the same input data. In addition, the Kalman version of the ideal algorithm (as presented in Section III-B) is run with the same tolerances as in the particle version.

First, the estimates produced by these two algorithms are compared with exact values computed offline by the disturbance smoother. The results are displayed in Figure 2 from which it is clear that estimates improve with decreasing tolerance. For \( \varepsilon = 0.5 \), the estimates exhibit clear fluctuations around the true values while for \( \varepsilon = 10^{-3} \), the estimates close smoothly to the true quantities. Figure 3 reports numerical values of the mean squared errors (MSEs). Moreover, Figure 4 displays the variance estimates \( \hat{\sigma}_s^2 \) for different values of \( s \), and the plot is well in line with the exponentially decreasing bound provided by Theorem 2. Finally, we study the truncation lags \( s^N(t) - s \) determined by the algorithm for different tolerances. The average lags across the 100 runs are displayed in Figure 4 from which it is evident that decreasing \( \varepsilon \) leads, in accordance with Theorem 3 on the average to larger lags. The slope in the end of the plot indicates non-truncation.
B. Stochastic Volatility Model

The second model of consideration is the stochastic volatility model

\[ X_{t+1} = \phi X_t + \sigma U_{t+1}, \]
\[ Y_t = \beta \exp(\frac{X_t}{2}) V_t, \]

where \( \{U_t\}_{t \in \mathbb{N}} \) and \( \{V_t\}_{t \in \mathbb{N}} \) are independent sequences of mutually independent standard Gaussian noise variables, and \( X_0 \sim \mathcal{N}(0, \sigma^2/(1-\phi^2)) \). In this nonlinear SSM, \( \{Y_t\}_{t \in \mathbb{N}} \) can be thought of as the log-returns of a stock while \( \{X_t\}_{t \in \mathbb{N}} \) can then be thought of as the unobserved log-volatility of the observed returns.

As before, we simulate 201 observations from the model indexed by \((\phi, \sigma, \beta) = (0.98, \sqrt{3}, \sqrt{7})\). We employ the adaptive-lag smoother targeting again the mean of the marginal smoothing distribution, i.e., \( h_s = \text{id} \) for all \( s \in \mathbb{N} \). Algorithm \ref{alg:adap-lag} is run with the tolerances \( \varepsilon \in \{0.5, 1, 10^{-3}\} \) and sample sizes \((N, \tilde{N}) = (400, 2)\). The algorithm is executed 200 times for each value of \( \varepsilon \), and all runs were based on the same data. Since exact computation is infeasible for this nonlinear model, we used proxies for the true quantities obtained as the averages of 10 independent replicates of the full PaRIS, without stopping criterion and using \((N, \tilde{N}) = (1000, 2)\).

In Figure \ref{fig:stoch-vol-smooth} we present the results of our runs. As can be seen for \( \varepsilon = 0.5 \), the marginal smoothing estimates of the adaptive-lag smoother deviate significantly from the ground truth of the PaRIS algorithm for most time-steps. Decreasing the tolerance to \( \varepsilon = 0.1 \) yields clearly improved—but still undesirably volatile—estimates. However, decreasing the tolerance even further to \( \varepsilon = 10^{-3} \) leads to a plot where the PaRIS estimates are firmly in the area of the estimates produced by the adaptive-lag smoother, and taking the mean over all the adaptive-lag estimates yields values indistinguishable from the estimates delivered by the PaRIS.

Finally, Figure \ref{fig:stoch-vol-lag} provides the average lags at different time steps, and obviously the nonlinear components of the model leads to a higher degree of adaptation compared to the linear Gaussian model.
VII. CONCLUSION

In this paper we have presented a novel algorithm—an adaptive-lag smoother—for online computation of marginal-smoothing expectations in general SSMs. We are not aware of any other algorithm in the literature solving satisfactorily this challenging problem. The proposed algorithm differs from standard fixed-lag smoothing [16], [19] in essentially two ways: first, the estimators produced by the algorithm do not at all suffer from particle path degeneracy; second, the lag is chosen adaptively in the algorithm by means of a variance criterion. The algorithm is furnished with a few theoretical results, including the asymptotic limit of the algorithm and a time uniform bound on the asymptotic memory requirement. Finally, the efficiency of the algorithm is illustrated numerically.

A result that is missing in the present paper is a rigorous theoretical analysis of the estimator’s bias and its dependence on the tolerance \( \varepsilon \). Since the estimator is driven by ergodicity arguments, such an analysis requires control of the forgetting of the backward chain. However, this is still an open—and presumably very complex—problem, and the question is thus left as future research.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore. Optimal Filtering. Prentice-Hall, 1979.
[2] O. Cappe. Online EM algorithm for hidden Markov models. Journal of Computational and Graphical Statistics, 20(3):728–749, 2011.
[3] O. Cappe, E. Moulines, and T. Rydén. Inference in Hidden Markov Models. Springer, 2005.
[4] S. Chib, F. Nadari, and N. Shephard. Particle Markov chain Monte Carlo methods for stochastic volatility models. J. Econometrics, 108:281–316, 2002.
[5] Roman Holenstein Christophe Andrieu, Arnaud Doucet. Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(3):269–342, 2010.
[6] P. De Jong. A cross validation filter for time series models. Stat. Comput., 10:197–208, 2000.
[7] P. Del Moral. Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications. Springer, 2004.
[8] P. Del Moral, A. Doucet, and S. Singh. A Backward Particle Interpretation of Feynman-Kac Formulae. ESAIM M2AN, 44(5):947–975, 2010.
[9] P. Del Moral, A. Doucet, and S. Singh. Forward smoothing using sequential Monte Carlo. Technical report, Cambridge University, 2010.
[10] R. Douc, A. Garivier, E. Moulines, and J. Olsson. Sequential Monte Carlo smoothing for general state space hidden Markov models. Ann. Appl. Probab., 21(6):2109–2145, 2011.
[11] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte-Carlo sampling methods for Bayesian filtering. Stat. Comput., 10:197–208, 2000.
[12] S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for non-linear time series. J. Am. Statist. Assoc., 50:438–449, 2004.
[13] J. Hull and A. White. The pricing of options on assets with stochastic volatilities. J. Finance, 42:281–300, 1987.
[14] P. E. Jacob, L. M. Murray, and S. Rubenthaler. Path storage in the particle filter. Statistics and Computing, pages 1–10, 2013.
[15] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, 1997.
[16] G. Kitagawa and S. Sato. Monte Carlo smoothing and self-organising state-space model. In Sequential Monte Carlo methods in practice, Stat. Eng. Inf. Sci., pages 177–195. Springer, New York, 2001.
[17] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. N. Canagarajah, and B. Ristic. Mobile tracking in cellular networks using particle filtering. IEEE Transactions on Wireless Communications, 6(10):3589–3599, 2007.

APPENDIX

We preface the proofs by some additional notation. For all \( t \in \mathbb{N} \), define the unnormalised transition kernels

\[
L_t(x_t, dx_{t+1}) := g_{t+1}(x_{t+1}) Q(x_t, dx_{t+1}),
\]

with the convention that \( L_t \equiv 1 \) when \( s > t \). In addition, we let \( L_{-1}(x, dx_0) := \delta_0(x) \delta_d(dx_0) \). Moreover, we may express each joint smoothing distribution \( \tilde{P}_{0:t|t} \) as \( \phi_t T_t \), where we have defined the kernels

\[
T_t(x_t, dx_{t-1}) := \begin{cases} 
\prod_{s=0}^{t-1} \tilde{Q}_{s+1} x_{s+1}, dx_s \\
\text{id} 
\end{cases} 
\]

for \( t \in \mathbb{N}^* \), \( t = 0 \).

Notice that \( T_{s|t}(x_s) \) in (10) can be expressed as \( T_{s|t}(x_s) = T_t h_s(x_s) \). Finally, define the operator

\[
D_t : h \mapsto T_t (h - \phi_{0:t|t} h),
\]

acting on the space of bounded measurable functions.

Proof of Lemma 7 By [21] Lemma 13 it holds that

\[
\sum_{i=1}^N \frac{\omega_i^t}{\Omega_t} (\tau_{s|i})^2 \overset{p}{\to} \phi_t (T_t^2 h_s) + \eta_t,
\]

where

\[
\eta_t := \sum_{s=t}^{t-1} \bar{N}_{t-\ell} \phi_t L_t \{ \tilde{Q} \delta_{T_t h_s - T_{t+1} h_s^2} L_{t+1} \cdots L_{t-1} \}.
\]

In addition, from [21] Theorem 1 we get that

\[
\sigma_{s|t}^2 \overset{p}{\to} \phi_t^2 (T_t h_s),
\]

and combining the previous two limits yields, as \( N \to \infty \),

\[
\sigma_{s|t}^2 = \sum_{i=1}^N \frac{\omega_i^t}{\Omega_t} (\tau_{s|i})^2 - \left( \sum_{i=1}^N \frac{\omega_i^t}{\Omega_t} \tau_{s|i} \right)^2 \overset{p}{\to} \sigma_{s|t}^2 := \phi_t (T_t^2 h_s) - \phi_t^2 (T_t h_s) + \eta_t = \phi_t \{ (T_t h_s - \phi_t T_t h_s)^2 \} + \eta_t.
\]
Thus, by [10] Lemma 10 it holds, for all \( t \geq s \),
\[
\|D_t^2 h_s\|_\infty \leq g^{t-s}|h|_\infty,
\]
giving us the bound
\[
\|\varphi (D_t^2 h_s)\|_\infty \leq g^{2(t-s)}|h|_\infty^2
\]
on the first term of (11). In order to bound the second term of (11), i.e. \( \eta_t \), we note that, [10] Lemma 10 yields that
\[
\|T_t h_s - T_{t+1} h_s\|_\infty \leq 2 g^{t-s}|h|_\infty.
\]
In addition, under Assumption 2, for all \( x \in X \),
\[
\varepsilon \mu(g_t L_{t+1} \cdots L_{t-1} I_1) \leq L_{t+1} \cdots L_{t-1} \{ x(x) \leq \varepsilon \mu(g_{t+2} L_{t+1} \cdots L_{t-1} I_1). \]
Combining the previous two bounds allows \( \eta_t \) to be bounded; indeed, proceed like
\[
\frac{\phi_t L_t}{Q_\varphi (T_t h_s - T_{t+1} h_s) L_{t+1} \cdots L_{t-1} I_1} \leq 4 g^{2(t-s)}|h|_\infty^2 = g^{2(t-s)} \phi_t^2. \]
Assuming now that \( \tilde{N} g^2 \neq 1 \) we may bound \( \eta_t \) using that
\[
\eta_t \leq 4 \frac{|h|_\infty^2}{1 - \tilde{N} g^2} \sum_{s=1}^{t-1} \tilde{N}^{t-s-1} g^{2(t-s)}
= 4 \frac{|h|_\infty^2}{1 - \tilde{N} g^2} \tilde{N}^{t-2} \sum_{s=1}^{t-1} (\tilde{N} g^2)^\ell
= 4 \frac{|h|_\infty^2}{1 - \tilde{N} g^2} \left( \tilde{N}^{t-2} - 2 g^{2(t-s)} \right). \tag{12}
\]
On the other hand, if \( \tilde{N} g^2 = 1 \), (12) yields
\[
\eta_t \leq 4 \frac{|h|_\infty^2}{1 - \tilde{N} g^2} \tilde{N}^{-t-2} (t-s).
\]
The previous may be summarised as
\[
\sigma_{s,t}^{2,\infty} \leq |h|_\infty^2 \begin{cases} \frac{c_1}{2} g^{2(t-s)} + c_2 \tilde{N}^{(t-s)} & \text{if } \tilde{N} g^2 \neq 1, \\ g^{2(t-s)} + c_3 (t-s) \tilde{N}^{(t-s)} & \text{if } \tilde{N} g^2 = 1, \end{cases}
\]
where
\[
c_1 := 1 - \frac{4}{(1-\tilde{N} g^2)(1-\tilde{N} g^2)}, \quad c_2 := \frac{4}{(1-\tilde{N} g^2)(1-\tilde{N} g^2)}, \quad c_3 := \frac{4}{(1-\tilde{N} g^2)}. \]