Simulation Smoothing: an Extremum Monte Carlo Approach

K. Moussa*

Vrije Universiteit Amsterdam
Department of Econometrics and Data Science

March 3, 2024

Abstract

This paper introduces a novel simulation smoothing method for state space models. The method can be used to compute smoothed estimates of the states and nonlinear functions of the states, and it allows for visualization of the joint smoothing distribution. The simulation smoother is based on the *extremum Monte Carlo* method. It uses simulated data from the model to estimate the conditional density functions from the backward decomposition of the joint smoothing density. The approach is generally applicable and deals naturally with missing data, as well as measurements that become available at mixed frequencies. The method is illustrated via examples with missing data, multimodal distributions, and intractable model densities. The flexibility and computational efficiency of the approach is demonstrated in an empirical application to a time series of Bitcoin based on the stochastic volatility model with stable errors.

Keywords: Fixed-interval smoothing, Hidden Markov model, Nonlinear non-Gaussian state space model, Stable distribution.

*E-mail: k.moussa@vu.nl. This paper is based on Chapter 4 of my PhD dissertation (Moussa, 2024). I am grateful for the comments from participants of the 2023 Econometrics Brownbag Seminar at the Vrije Universiteit Amsterdam.

1 Introduction

The estimation of variables that are subject to measurement noise has a long history in statistics. From Galileo's random observational errors (Galilei, 1632; Hald, 1986) to the least squares method of Legendre (1805) and Gauss (1809), the task stands at the origin of the field as a scientific discipline, and it has given rise to many new strands of the statistical literature with related problems. A well-known example is that of *smoothing*, in which a sequence of latent variables is estimated from corresponding noisy measurements based on all the data that are available ex post. The smoothing task is a long-standing problem, dating back at least to the pioneering work of Whittaker (1923) on posterior mode estimation under smoothness constraints.

If the measurements are ordered by time, the standard approach to smoothing is based on state space models (SSMs); see Durbin and Koopman (2012) and Chopin and Papaspiliopoulos (2020). To fix notation, consider the following formulation of the SSM,

$$\begin{aligned} y_t &= m_t(x_t, \varepsilon_t^y), & (\varepsilon_t^x, \varepsilon_t^y) &\sim p(\varepsilon_t^x, \varepsilon_t^y), \\ x_{t+1} &= s_t(x_t, \varepsilon_t^x), & x_1 &\sim p(x_1), \end{aligned} \tag{1}$$

for $t = 1, \ldots, T$, with $x_t \in \mathbb{R}^{N_x}$ the unobserved state vector at time t, $y_t \in \mathbb{R}^{N_y}$ the corresponding vector of measurements (or observations), noise vectors ε_t^x and ε_t^y, and $N_x, N_y, T \in \mathbb{N}$, where T is the length of the time series. In addition, $p(\cdot)$ denotes the probability density function (PDF) or probability mass function (PMF) of its argument. In terms of the above SSM formulation, (fixed-interval) smoothing amounts to conducting inference on the unobserved states x_t, $t = 1, \ldots, T$, conditional on all measurements $y_{1:T} = \{y_t\}_{t=1}^T$. A common approach is to use *simulation smoothing*, that is, drawing paths of the states from the joint smoothing density,

$$x_{1:T} \sim p(x_{1:T}|y_{1:T}). \tag{2}$$

The task of simulation smoothing is important for several reasons. First, the simulated paths allow for computing estimates of the states and functions of the states, $g(x_{1:T})$, conditional on the data. For example, by choosing g appropriately, the paths can be used to estimate the posterior means, variances, and other moments of the states. Second, the ability to sample from the above posterior density is crucial to Bayesian inference in dynamic models with latent variables; see Jacquier, Polson, and Rossi (1994) and Kim, Shephard, and Chib (1998) for applications in the context of stochastic volatility (SV) models. In a classical setting, drawing paths with high posterior density values is needed to ensure that simulated maximum likelihood procedures remain computationally feasible. For SSMs, the likelihood function is an integral with respect to the latent states, $p(y_{1:T}) = \int p(x_{1:T}, y_{1:T}) dx_{1:T}$, which can be evaluated by Monte Carlo integration based on simulated state paths. Since paths with low values of the joint density contribute little to the integral, it is preferable to use importance sampling by drawing from the joint smoothing density as in (2) or, if this is not possible, from a close approximation thereof (Durbin & Koopman, 1997; Shephard & Pitt, 1997). Third, the simulated paths enable investigating the joint behavior of past states conditional on the data, and they allow for visualization of the joint smoothing distribution. As noted by Godsill, Doucet, and West (2004), "*Generating sample realizations is the most efficient, effective, and intuitive approach to studying complicated multivariate joint distributions.*"

To perform simulation smoothing, one can exploit the following backward decomposition of the joint smoothing density (Carter & Kohn, 1994; Frühwirth-Schnatter, 1994),

$$p(x_{1:T}|y_{1:T}) = p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1:T}, y_{1:T}) \tag{3}$$

$$= p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1}, y_{1:T}) \tag{4}$$

$$= p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1}, y_{1:t}), \tag{5}$$

where (3) follows from Bayes' formula, (4) holds when the states follow a Markov process, and (5) holds under the additional assumptions that the measurement noise is serially independent and only contemporaneous dependence between the noise vectors is allowed ($\varepsilon_s^x \perp \varepsilon_t^y \ \forall \ s \neq t$). The above assumptions are standard in state space modeling. By performing draws from the densities p in (5), simulation smoothing can be performed via the backward sampling procedure in Algorithm 1. This approach is directly applicable when the SSM is linear and Gaussian, which holds if the measurement and state transition functions m_t and s_t in (1) are linear, and all densities p are Gaussian. The linear Gaussian case is discussed by Carter and Kohn (1994) and Frühwirth-Schnatter (1994), as well as De Jong and Shephard (1995) and Durbin and Koopman (2002), who propose related simulation smoothers defined in terms of the noise vectors.

For nonlinear and/or non-Gaussian SSMs, which are pervasive in practice (Doucet, De Freitas, & Gordon, 2001), the above approach is usually not feasible. In such case, simulation smoothing is more challenging and is often performed via Markov chain Monte Carlo (MCMC) or sequential Monte Carlo (SMC) methods for smoothing. The former method has as main drawback that it often requires the candidate density to be tailored to the problem to work well; see the above-mentioned references on SV models for examples. SMC methods for smoothing, on the other hand, become computationally expensive when it is needed to draw many paths for accurate inference. For example, the simulation smoothing method of Godsill et al. (2004) has complexity $O(TND)$ for D draws with N particles, while most SMC smoothing methods have a complexity of $O(N^2)$; see Chopin and Papaspiliopoulos (2020, Ch. 12) for a recent overview of SMC smoothing

Algorithm 1 Backward sampling method for simulation smoothing.

To draw D paths $x_{1:T}^{[j]},\ j = 1, \ldots, D$, from the joint smoothing density $p(x_{1:T}|y_{1:T})$:

1. **Draw states at time** T.
   For $j = 1, \ldots, D$:
   Draw $x_T^{[j]} \sim p(x_T|y_{1:T})$.

2. **Draw remaining states moving backwards in time**.
   For $t = T-1, \ldots, 1$:
   For $j = 1, \ldots, D$:
   Draw $x_t^{[j]} \sim p(x_t|x_{t+1}^{[j]}, y_{1:t})$.

methods. Standard versions of MCMC and SMC also impose tractability requirements on the SSM that are not always met in practice. For instance, both methods require evaluating the measurement density $p(y_t|x_t)$ and state transition density $p(x_{t+1}|x_t)$, which may not be available in closed form. Some examples are: the stable and related distributions (Barndorff-Nielsen & Shephard, 2001), the majority of the wrapped distributions (Mardia, Jupp, & Mardia, 2000), and discretizations of non-Gaussian continuous-time processes, which are common in finance (Creal, 2008) and ecology (e.g., Chopin & Papaspiliopoulos, 2020, Ch. 2.4.5). In addition, most SMC methods for smoothing cannot handle the case in which these distributions are degenerate, such as when the state follows an autoregressive process of order higher than one; see Fearnhead, Wyncoll, and Tawn (2010) for a discussion of this point.

This paper introduces a novel simulation smoother based on *extremum Monte Carlo* (XMC; Blasques, Koopman, & Moussa, 2023a). The method uses simulated data from the SSM to estimate the conditional density functions p from the backward decomposition in (5). By using density estimators that allow for direct simulation, the backward sampling approach in Algorithm 1 can be used to perform simulation smoothing. The proposed method has several advantages. It is generally applicable, as its main requirement is the ability to simulate from the SSM. For example, the densities from the general SSM in (1) and the backward decomposition in (5) do not have to be analytically tractable. In addition, the assumptions on the dependency structure of the noise terms and the Markov assumption on the states can be dropped by considering the appropriate conditioning sets in (3) or (4). The method also deals naturally with missing data and measurements that become available at mixed frequencies. Moreover, one of the proposed versions of the simulation smoother has complexity $O\big(T(N+D)\big)$, which makes the approach suitable for applications that require both a precise approximation and a large number of draws.

The remainder of this paper is organized as follows. Section 2 introduces the XMC simulation smoothing method. Section 3 discusses several important properties of the method by means of numerical examples. Section 4 presents an empirical application of volatility smoothing for a time series of Bitcoin. Section 5 concludes.

2 Simulation smoothing

2.1 The simulation smoothing algorithm

Algorithm 2 presents the XMC method for simulation smoothing. The algorithm takes as input an instance of the SSM in (1), a set $\mathbb{F}_N$ of conditional densities (or PMFs), and a corresponding loss function L to generate as output D paths from the estimated smoothing distribution. This is done in three steps: simulation, fitting, and backward sampling.

The algorithm starts by using the SSM to simulate N paths of the states $x \coloneqq x_{1:T}$ and observations $y \coloneqq y_{1:T}$, where we will omit the subscripts of the full paths for conciseness. The simulated data are split into training and validation samples, with the latter used to determine the optimal tuning parameters of the conditional density estimator. In the optimization in (6), Y_t denotes the conditioning set at time t from the backward decomposition in (5), and $\widetilde{Y}_t \subseteq Y_t$ denotes a corresponding subset of covariates that is used for regularization. More specifically, the covariate set is defined in terms of a window

Algorithm 2 Extremum Monte Carlo method for simulation smoothing.

1. **Simulate**: Use the SSM in (1) to simulate N paths of the states and observations,

$$x^{(i)},\ y^{(i)}, \qquad i = 1, \ldots, N,$$

where $x \coloneqq x_{1:T}$ and $y \coloneqq y_{1:T}$.

2. **Fit**:

   (a) *Split data*: Set $c_{\text{val}} \in (0, 1)$ and split the data into training and validation samples with sizes

$$N_{\text{tr}} = N - N_{\text{val}} \qquad \text{and} \qquad N_{\text{val}} = [c_{\text{val}} N].$$

   (b) *Regularization*: For a set of candidate tuning parameters, perform the following extremum estimation at times $t = T,\ T - 1$:

$$\widehat{f}_t^N \in \underset{f \in \mathbb{F}_N}{\arg\min} \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} L\left(x_t^{(i)}, \widetilde{Y}_t^{(i)}, f\right), \tag{6}$$

   with covariates $\widetilde{Y}_t^{(i)} \subseteq Y_t^{(i)}$, where Y_t denotes the conditioning set at time t from the backward decomposition in (5), and with $\mathbb{F}_N$ a set of conditional densities $f(x_t|\widetilde{Y}_t)$ to estimate $p(x_t|Y_t)$. Determine the optimal tuning parameters by minimizing the loss for a separate validation sample.

   (c) *Estimate*: Use the regularized tuning parameters to perform the estimation in (6) at all times $t = T, \ldots, 1$ to obtain the function estimates $\{\widehat{f}_t^N\}_{t=1}^T$.

3. **Backward sampling**: Apply backward sampling via Algorithm 1 using the estimated densities $\widehat{f}_t^N(x_t|\widetilde{Y}_t)$ evaluated at the observed data to draw D paths:

$$x^{[j]} \sim \prod_{t=1}^{T} \widehat{f}_t^N(x_t|\widetilde{Y}_t) \approx p(x|y), \qquad j = 1, \ldots, D.$$

size parameter, $W \in \{1, \ldots, T\}$, to contain the W observations nearest to time t:

$$\widetilde{Y}_t = \begin{cases} y_{\underset{\sim}{T}:T} & \text{if } t = T, \\ y_{\underset{\sim}{t}:t} \cup x_{t+1} & \text{if } t < T, \end{cases} \qquad \underset{\sim}{t} = \max\{t - W + 1, 1\}, \tag{7}$$

for $t = T, \ldots, 1$. Together with the tuning parameters that are specific to the estimator $\widehat{f}_t^N$, the window size W is determined in the regularization step. This consists of selecting the minimizer of the validation loss from several candidate tuning parameters generated by a Bayesian optimization procedure (Bergstra, Yamins, & Cox, 2013). To save computations, the regularization step is performed only at times T and $T - 1$. The tuning parameters determined at time $T - 1$ are re-used for the other times $t < T - 1$, as the corresponding minimization problems in (6) are similar. Further computational savings could be obtained by re-using the density estimates to perform simulation at other time points; this extension to Algorithm 2 is discussed in the next section. At time T, however,

there is no future state to condition on, hence it can be expected that this case requires a larger window size. In the estimation step, the regularized tuning parameters are used to perform the estimation for all times $t = T, \ldots, 1$. In the backward sampling step, the conditional density estimators are evaluated at the observed data, and the resulting densities are used to draw D smoothed paths via Algorithm 1 with $p(x_t|Y_t) \coloneqq \widehat{f}_t^N(x_t|\widetilde{Y}_t)$ for $t = T, \ldots, 1$.

For each choice of density estimator, Algorithm 2 defines a corresponding XMC simulation smoother. The optimal choice of estimator will depend on the instance of the SSM. For broad applicability of the method, we require that the estimator is general enough to provide a good approximation to most smoothing densities of practical interest. Further requirements are that the estimator should be able to deal with many covariates (for large W or N_y), and performing draws from the estimated distributions must be inexpensive to ensure that the simulation smoother is computationally efficient. The method will therefore be illustrated with the following two widely-used estimators: the mixture density network (MDN; Bishop, 1994) and the quantile regression forest (QRF; Meinshausen, 2006). The MDN is a mixture of normal densities in which the parameters are defined as the output of a neural network. The universal approximation properties of neural networks and mixtures of normal densities ensure that this estimator is generally applicable (Bishop, 1994). The QRF provides a discrete approximation to the target densities. This method uses the fact that the prediction of a random forest can be represented as a convex combination of the realizations of the dependent variable. The corresponding weights are used to define an estimate of the conditional cumulative distribution function. A brief discussion of both methods can be found in Appendix A.

We now discuss several computational considerations regarding Algorithm 2. First, the algorithm is highly amenable to parallelization, as the minimizations in (6) can be performed simultaneously for different times t. This allows for decreasing the runtime of the estimation step by increasing the number of physical cores. Regarding storage, it may be convenient to save the function estimates $\widehat{f}_t^N$ for re-use, which is convenient for repeated smoothing tasks in which the static parameters remain constant (or approximately so). On the other hand, if memory is scarce, the draw step can be performed immediately after the estimation step for each time t, so that the function estimates can be discarded directly after the draw.

Table 1 shows the computational complexity for the two XMC simulation smoothers considered. For the QRF estimator, the resampling step has a worst-case complexity of $O(ND)$. Because the MDN estimator is parametric (for given tuning parameters), the costs of the draw step do not directly depend on N, which results in an overall complexity of $O\big(T(N + D)\big)$. This gives a crucial improvement over the QRF version, as well as the method of Godsill et al. (2004), which has complexity $O(TND)$ for N

Table 1: Computational complexity for the mixture density network (MDN) and quantile regresion forest (QRF) methods and corresponding XMC simulation smoothers.

Method	Estimation	Draw	Simulation smoother
MDN	$O(N)$	$O(D)$	$O\big(T(N + D)\big)$
QRF	$O\big(N \log(N)\big)$	$O(ND)$	$O\big(TN(\log(N) + D)\big)$

particles. It is important to note that the computational costs of drawing (unconditional) paths are lower—often substantially—than those of drawing particles, as SMC methods also contain a resampling step, and the use of an importance sampler usually leads to a further increase of the computational costs.

2.2 Steady state simulation smoothing

The similarity between the optimization problems in (6) for times $t < T$ can be exploited to save computations, which is especially relevant when dealing with long time series. This section introduces a *steady state* (SS) extension to Algorithm 2, which re-uses the function estimate at some appropriate time t_{ss} at other times $t < t_{\text{ss}}$ to circumvent the corresponding estimation steps.[1] A minimal requirement for this approach to be sensible is that the respective covariate sets are translations of each other, such that the covariate set at time t is the result of applying $(t_{\text{ss}} - t)$ times the lag operator to every element in $\widetilde{Y}_{t_{\text{ss}}}$, the covariate set at time t_{ss}. We thus have that

$$\widetilde{Y}_t = \left\{ z_{j-(t_{\text{ss}}-t)} \middle| z_j \in \widetilde{Y}_{t_{\text{ss}}} \right\}, \tag{8}$$

where the covariate sets are defined via (7), so the elements z_j either correspond to the next state, $x_{t_{\text{ss}}+1}$, or to an observation from $\{y_1, y_2, \ldots, y_{t_{\text{ss}}}\}$. This implies the feasible range

$$W \leq t \leq T - 1,$$

where the upper bound on t reflects the availability of a future state for conditioning, while the lower bound ensures that the observations used, $y_{\underset{\sim}{t:t}}$, are consistent with (8) because for $t \geq W$ we have

$$\underset{\sim}{t+1} = \max\{(t+1) - W + 1, 1\} = \max\{t - W + 1, 1\} + 1 = \underset{\sim}{t} + 1.$$

To apply the SS approach, we check when performing the estimation step for $t = T-1, \ldots, W+1$ if

$$\sum_{i=1}^{N_{\text{val}}} L\left(x_W^{\langle i \rangle}, \widetilde{Y}_W^{\langle i \rangle}, \widehat{f}_t^N\right) \leq (1 + c_{\text{ss}}) \sum_{i=1}^{N_{\text{val}}} L\left(x_W^{\langle i \rangle}, \widetilde{Y}_W^{\langle i \rangle}, \widehat{f}_W^N\right),$$

with superscripts $\langle i \rangle$ indicating cases from the validation sample $(x^{\langle i \rangle}, y^{\langle i \rangle})$, $i = 1, \ldots, N_{\text{val}}$, and $c_{\text{ss}} \geq 0$ a chosen tolerance level. The validation loss is compared at time $t = W$, the last estimation time in the feasible range, to estimate the largest increase in the loss, since the $\widehat{f}_t^N$ are expected to differ more the further in time they are apart. If the condition in (2.2) is satisfied at time t, we say that an SS has been reached and set $t_{\text{ss}} \coloneqq t$, after which the estimate $\widehat{f}_{t_{\text{ss}}}^N$ can be used to circumvent the estimations at the remaining times in the feasible range, $W \leq t < t_{\text{ss}}$

[1] See Blasques et al. (2023a) for a similar idea in the context of filtering.

2.3 Inference by importance sampling

Algorithm 2 relies on function estimators, which suggests two ways in which the corresponding inference can become exact. The first is direct, in the sense that each estimator converges to the corresponding density from the backward decomposition as the number of simulated paths N diverges to infinity ("N-convergence"). The corresponding analysis is given in Appendix B. The second approach makes use of importance sampling (Durbin & Koopman, 2012, Ch. 11) and considers convergence as the number of draws D diverges to infinity ("D-convergence"). This approach utilizes several well-known results from the importance sampling literature and is discussed below.

For some fixed $N \in \mathbb{N}$, denote the XMC estimator of the joint smoothing density by

$$\widehat{f}(x|y) = \prod_{t=1}^{T} \widehat{f}_t^N(x_t|\widetilde{Y}_t^N).$$

If $p(x, y)$ can be evaluated, then importance sampling is based on the identity

$$\mathbb{E}[g(x)|y] = \int g(x)p(x|y)dx = \int g(x)\frac{p(x|y)}{\widehat{f}(x|y)}\widehat{f}(x|y)dx \propto \int g(x)\frac{p(x,y)}{\widehat{f}(x|y)}\widehat{f}(x|y)dx, \quad (9)$$

which shows that the conditional expectation of any function g of the states path x can be evaluated by using an importance density $\widehat{f}(x|y)$ which contains the support of the target density $p(x|y)$. If, in addition to the expectation in (9), the marginal likelihood exists, $p(y) = \int p(x, y)dx \in \mathbb{R}$, then it follows from Theorem 1 in Geweke (1989) that the following estimator based on D independent draws from the importance density is consistent,

$$\sum_{i=1}^{D} \omega_i \, g(x^{[i]}) \xrightarrow{\text{a.s.}} \mathbb{E}[g(x)|y] \qquad \text{as} \qquad D \to \infty, \quad (10)$$

with normalized importance weights ω_i given by

$$\omega_i = \frac{\widetilde{\omega}_i}{\sum_{j=1}^{D} \widetilde{\omega}_j}, \qquad \text{with} \qquad \widetilde{\omega}_i = \frac{p(x^{[i]}, y)}{\widehat{f}(x^{[i]}|y)}. \quad (11)$$

In the common case where the noise terms are mutually and serially independent, as well as independent from x_1, the weights are determined via the simple expression

$$p(x, y) = p(x_1)\prod_{t=2}^{T} p(x_t|x_{t-1})\prod_{t=1}^{T} p(y_t|x_t).$$

Besides the computation of moments, the importance weights can also be used to resample the draws with corresponding probabilities ω_i, which is called sampling importance resampling (SIR; Rubin, 1987). By letting $D \to \infty$, we thus obtain draws from $p(x|y)$ under the same assumptions that were needed to establish the convergence in (10); see Smith and Gelfand (1992). The SIR method was also used by Kim et al. (1998) for a similar purpose as in this paper, but in the context of Gibbs sampling.

2.4 Related literature

The simulation smoother builds upon the XMC filtering method introduced in Blasques et al. (2023a), which combines simulated data and regression (or "extremum estimation"; Amemiya, 1985) to estimate the filtering means $\mathbb{E}[x_t|y_{1:t}]$, quantiles, and modes for SSMs. In this work, however, we focus on the simulation of paths from the joint smoothing density, which allows for inference on the states that goes beyond mere point estimates.

The SS extension from Section 2.2 links the simulation smoother to the literature on amortized inference (e.g., Stuhlmüller, Taylor, & Goodman, 2013), in which Bayesian inference is performed by estimating posterior densities from simulated data; by re-using the estimated density function for multiple values of the data, the estimation costs are amortized. Two approaches related to the present work are due to Paige and Wood (2016), who consider amortized inference for Bayesian networks, and Lin and Eisner (2018), where the focus is on natural language processing. Besides the different setting, our method diverges from these approaches in several important ways. First, their aim is to find adequate importance samplers for SMC, whereas our approach to inference is either direct or via the importance sampling approach from Section 2.3. Second, neither reference discusses the crucial task of data reduction, which makes these approaches computationally infeasible for medium to long time series. Moreover, from a statistical learning point of view, the use of all available covariates ($\widetilde{Y}_t = Y_t$) is typically suboptimal. Lastly, both references focus exclusively on neural networks, which may not be the best method for the problem at hand. By contrast, Algorithm 2 is not linked to a specific estimation method, hence it can be combined with any conditional density or PMF estimator of choice.

3 Numerical examples

This section discusses several important properties of the XMC simulation smoother by means of numerical examples. Throughout, the noise terms ε_t^x and ε_t^y are assumed to be mutually and serially independent, as well as independent of the initial state x_1, and we set $c_{\text{val}} = 0.1$. In line with Bishop (1994), the MDN is based on a neural network with a single hidden layer.

3.1 A linear Gaussian test case

We consider the following univariate local level model,

$$\begin{aligned} y_t &= x_t + \varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{N}(0, \sigma_y^2), \\ x_{t+1} &= x_t + \varepsilon_t^x, & \varepsilon_t^x &\sim \mathrm{N}(0, \sigma_x^2), \end{aligned} \tag{12}$$

with initialization $x_1 \sim \mathrm{N}(\mu_1, \sigma_1^2)$ and parameters $\sigma_x, \sigma_y, \sigma_1 > 0,\ \mu_1 \in \mathbb{R}$. Since the above model is both linear and Gaussian, all densities p from the backward decomposition in (5) are also Gaussian (Durbin & Koopman, 2012, Lemma 1), so that the Kalman smoother (Anderson & Moore, 1979, Ch. 7) can be used to compute the means and quantiles of the marginal smoothing density $p(x_t|y)$ for $t = 1, \ldots, T$.

As a first test case for the proposed simulation smoother, we follow Durbin and Koopman (2012, Ch. 2) and apply the above local level model to the measurements of

the annual flow volume of the Nile river taken at Aswan from 1871 to 1970, which are shown in Figure 1 (the dots). The static parameters $\theta = (\mu_1, \sigma_1, \sigma_x, \sigma_y)'$ were set to the maximum likelihood estimates $\sigma_x = 38.329$ and $\sigma_y = 122.877$, with approximate diffuse initialization $\mu_1 = 0$ and $\sigma_1^2 = 10^7$. The XMC method was applied with $N = 10^5$ and $D = 10^4$. The first 10 paths for each version are shown in Figure 1, as well as the mean of the D paths. The sample mean of the MDN version coincides with the Kalman smoother, and for the QRF version this holds at most times. The paths are near the smoothing means computed by the Kalman smoother, which reflects the informativeness of the observations.

Figure 2 shows the 10%, 50%, and 90% quantiles of the marginal smoothing density $p(x_t|y)$ as computed by the Kalman smoother. Parts (b) and (c) show the corresponding quantiles computed by the MDN- and QRF-XMC simulation smoothers, respectively. As with the means, the MDN quantiles are seen to coincide, while the outer QRF quantiles are typically too wide. In this example, the good fit of the MDN version was to be expected: since all target densities are Gaussian, in principle, only a single component is needed in the mixture. In addition, the smoothing densities $p(x_t|y)$ are characterized by a conditional mean that is linear in the observations y_t and a variance that is independent of these elements (Durbin & Koopman, 2012, Lemma 1), so that a simple parameterization of the network suffices.

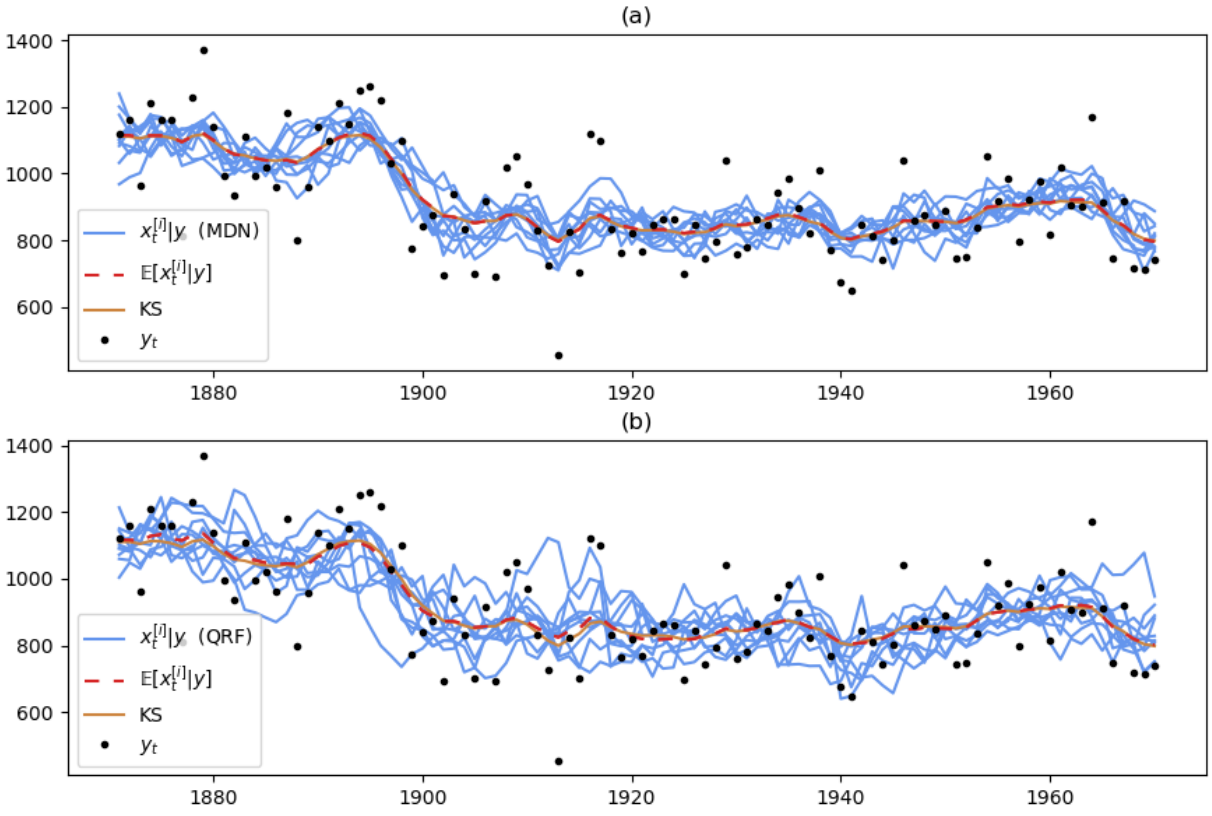

Figure 1: Paths from the XMC simulation smoother based on the local level model in (12) and the Nile data: (a) first 10 paths and means of the $D = 10^4$ paths for the mixture density network (MDN) XMC simulation smoother and the Kalman smoother (KS); (b) corresponding quantities for the quantile regression forest (QRF) version.

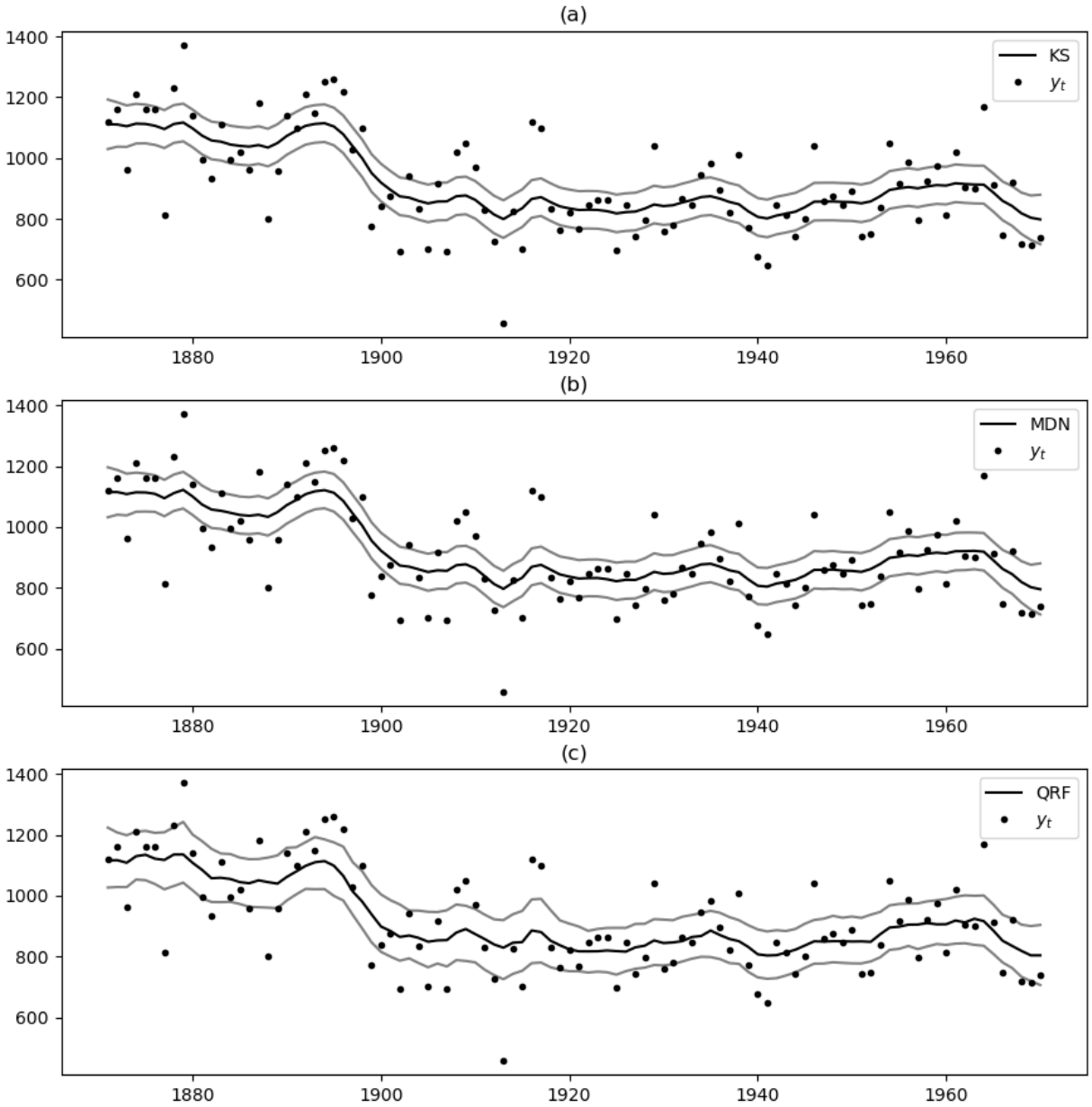

Figure 2: Quantiles of the marginal smoothing density $p(x_t|y)$ for cumulative probabilities 10%, 50%, and 90% and $t = 1, \ldots, T$ based on the local level model in (12) and the Nile data (dots): (a) Kalman smoother (KS); (b) mixture density network (MDN) XMC simulation smoother; (c) quantile regression forest (QRF) XMC simulation smoother.

3.2 Missing data

In practice it often occurs that some of the data are missing. The XMC method handles this situation simply by omitting the corresponding covariates when performing the estimations. To illustrate the approach, we return to the local level model illustration with the Nile data and treat the observations at times $t = 21, \ldots, 40$ and $t = 61, \ldots, 80$ as missing. Figure 3 shows the marginal 10%, 50%, and 90% smoothing quantiles based on the Kalman smoother and the MDN-XMC simulation smoother with $N = 10^5$ and $D = 10^4$. To ensure that sufficiently many observations were used despite the intervals

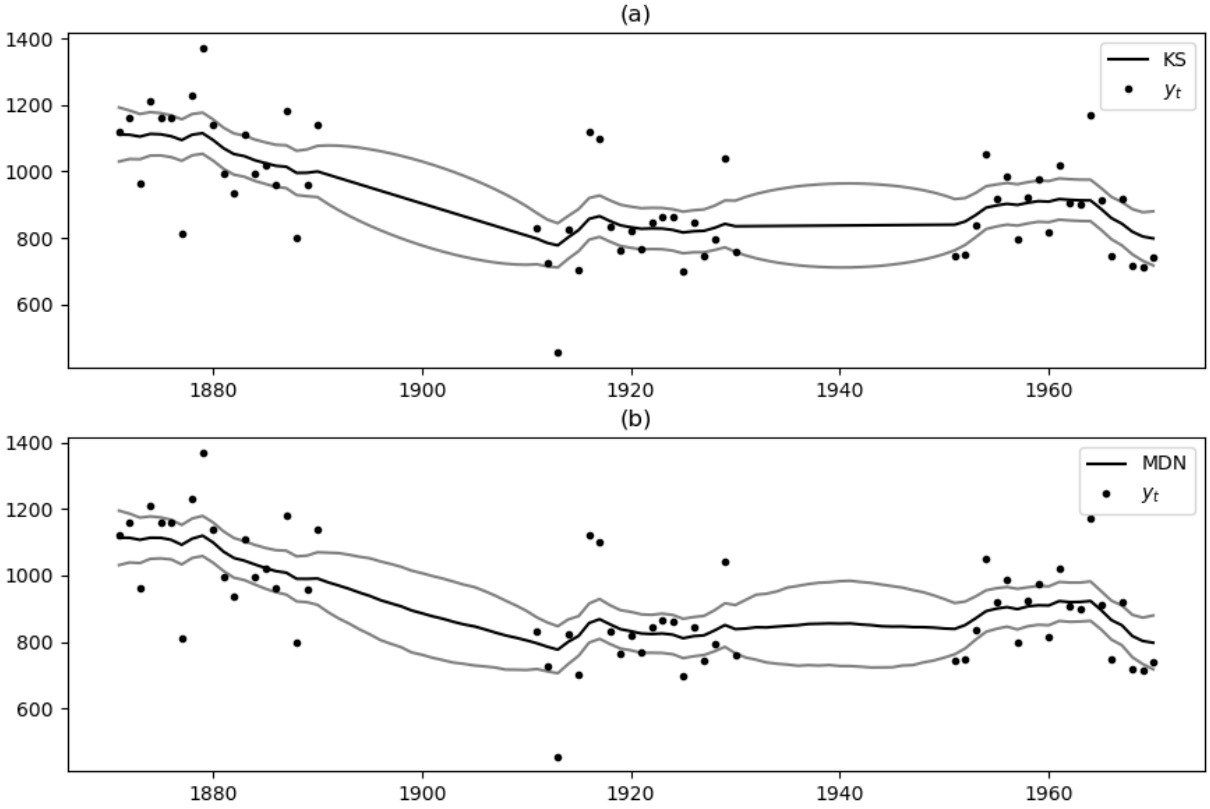

Figure 3: Marginal 10%, 50%, and 90% smoothing quantiles based on the local level model in (12) and the partial Nile data set, in which the observations at time points $21, \ldots, 40$ and $61, \ldots, 80$ are treated as missing: (a) Kalman smoother (KS); (b) mixture density network (MDN) XMC simulation smoother with $N = 10^5, D = 10^4$, and $W = 50$.

of missing data, the window size was set to $W = 50$. The quantiles based on the Kalman smoother and the XMC simulation smoother are again seen to be close to each other. In line with intuition, the quantiles widen in the missing data periods to reflect the increased uncertainty on the states.

The approach for handling missing data could also be used to deal with measurements that become available at mixed frequencies. A relevant application is simulation smoothing with mixed-frequency vector autoregressions, in which the observations are treated as states with those observed at lower frequencies treated as missing at an appropriate selection of the times; see Ankargren and Jonéus (2021) and the references therein.

3.3 Multimodal densities and visualization of the joint smoothing distribution

We consider simulation smoothing for the following nonlinear model (e.g., Gordon, Salmond, & Smith, 1993; Kitagawa, 1996),

$$\begin{aligned} y_t &= \frac{x_t^2}{20} + \varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{N}(0, \sigma_y^2), \\ x_{t+1} &= \frac{1}{2}x_t + \frac{25x_t}{1+x_t^2} + 8\cos\big(1.2(t+1)\big) + \varepsilon_t^x, & \varepsilon_t^x &\sim \mathrm{N}(0, \sigma_x^2), \end{aligned} \tag{13}$$

with $x_1 \sim \mathrm{N}(0, 1)$, and the static parameters are set to $\sigma_x^2 = 0.1$ and $\sigma_y^2 = 1$ as in Kitagawa (1996). A well-known property of this SSM is that many of its marginal densities are multimodal (Doucet et al., 2001, Ch. 1). This poses a challenge to some of the established methods for simulation smoothing. For instance, the common approach of using a Gaussian approximation (Durbin & Koopman, 1997; Shephard & Pitt, 1997) is not suitable because at least one mode of the target density would be neglected. Furthermore, it is well known that multimodal distributions pose difficulties for MCMC methods because the chain may get stuck at one of the modes, in which case it may take many iterations for the draws to become representative of the target distribution (e.g., Hoogerheide, Van Dijk, & Van Oest, 2009). The XMC simulation smoother does not face these issues because its estimators can be chosen general enough to accommodate multimodality, and the simulated paths are independent of each other.

For illustration, the nonlinear model was used to simulate a path of the states and observations shown in Figure 4. Using these observations, the MDN-XMC simulation smoother was applied with $N = 10^6$ and $D = 10^5$. Figure 4 (b) shows the resulting paths (the blue lines), which are depicted on top of an equal number of draws from the unconditional distribution of the states (the gray lines). Contrary to the unconditional paths, the smoothed paths are concentrated around the true states, which indicates that

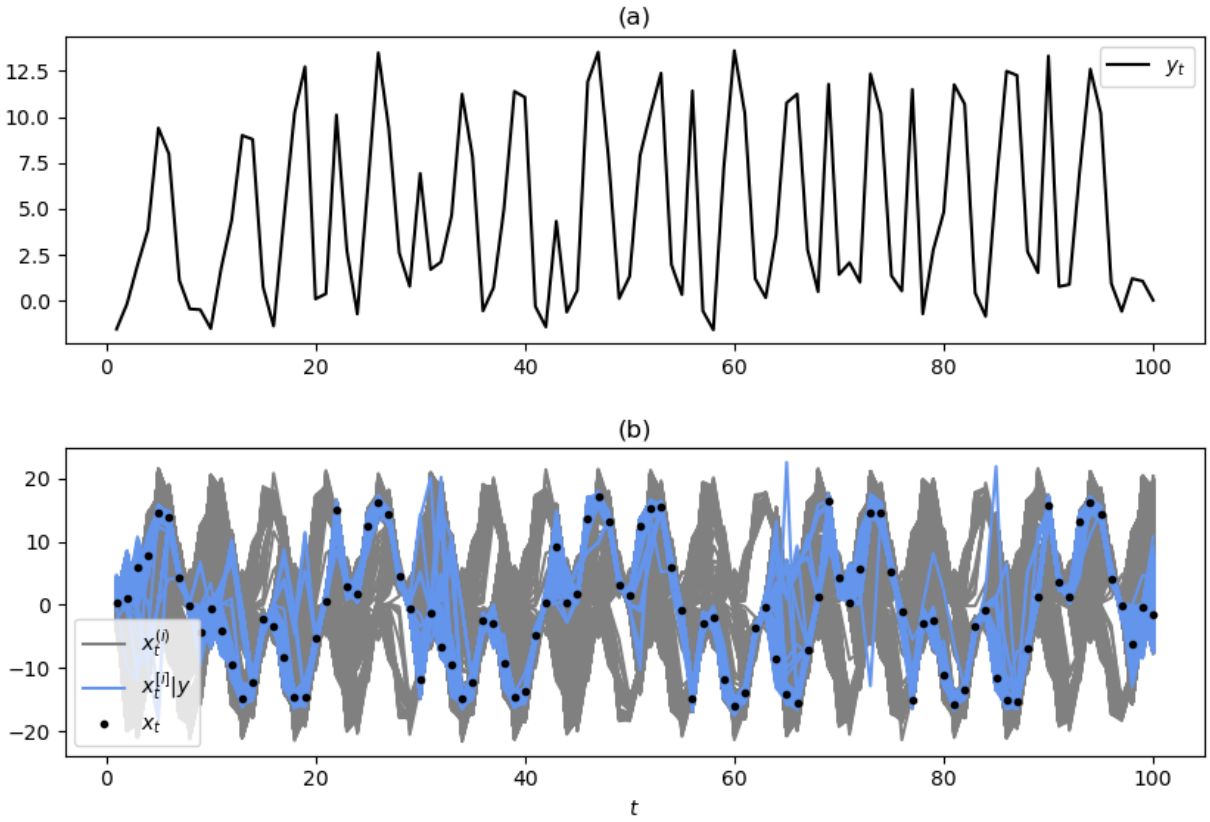

Figure 4: Simulated data and paths from the mixture density network XMC simulation smoother and the unconditional distribution of the states based on the nonlinear model in (13): (a) simulated path of the observations; (b) true states (dots) alongside $D = 10^5$ paths from the simulation smoother (the blue lines) and the unconditional distribution (the gray lines).

the observations are highly informative on the states.

At several times the paths suggest that the corresponding marginal distributions are multimodal, which is confirmed in Figure 5 for the marginal densities $p(x_t)$ and $p(x_t|y)$ with $t = 9$ and $t = 92$. The densities were estimated by a Gaussian kernel density estimator (KDE) based on the simulated paths. At $t = 9$ the smoothing density provides a drastic update of the locations of the modes relative to the unconditional density. At $t = 92$, conditioning on the observations yields a bimodal density, while the corresponding unconditional density is unimodal. This example illustrates the importance of smoothing for obtaining accurate estimates of the states and their corresponding marginals.

In addition to the direct XMC estimates, Figure 5 also shows the KDE based on the SIR technique from Section 2.3. In both cases the SIR correction is small, and it is pertains only to the height but not the location of the modes. This finding is illustrative for the other time points (not shown), which suggests that the XMC method provides a good approximation to the smoothing density.

An important advantage of simulation smoothers compared to regular smoothing methods is that the paths allow for an analysis and visualization of the *joint* smoothing distribution of the states. For example, Figure 6 shows scatter plots of the variates $x_{9:10}^{[i]}$ and $x_{92:93}^{[i]}$, which reveals that the joint smoothing distribution is a highly complex multimodal and non-elliptical distribution. The density $p(x_{9:10}|y)$ appears to be tri-modal and is clearly non-elliptical. The latter also holds for $p(x_{92:93}|y)$ shown in the right-hand side plot, which has a markedly different shape than $p(x_{9:10}|y)$. Similar non-Gaussian characteristics of the smoothing distribution are shown in Figure 7, which plots the contours of the KDEs $\widehat{f}(x_{14:15}|y)$ and $\widehat{f}(x_{31:32}|y)$. The use of contour (or surface) plots allows for a more detailed analysis of the joint smoothing distribution. For example, the plot of $\widehat{f}(x_{14:15}|y)$ indicates that there is a global mode is around $x_{14:15} = (-12.8, -2.7)$ and a local mode around $x_{14:15} = (-13.1, -3.5)$.

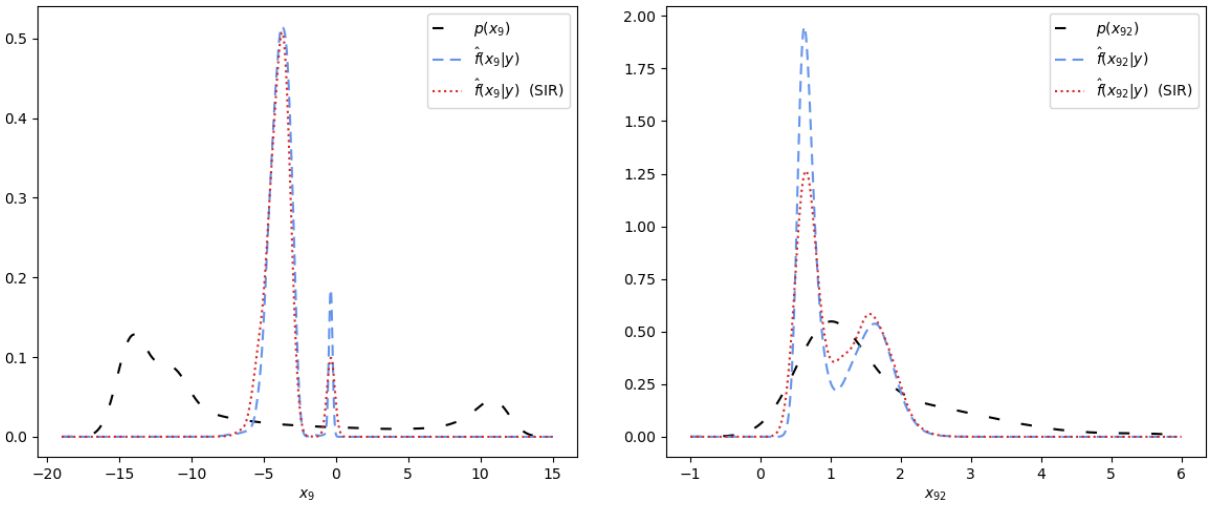

Figure 5: Marginal kernel density estimates at $t = 9$ and $t = 92$ based on the nonlinear model in (13) and the simulated measurements shown in Figure 4 (a). The estimates are shown for the unconditional density, the density based on the paths of the mixture density network XMC, and the density based on the latter paths with application of the sampling importance resampling (SIR) technique from Section 2.3.

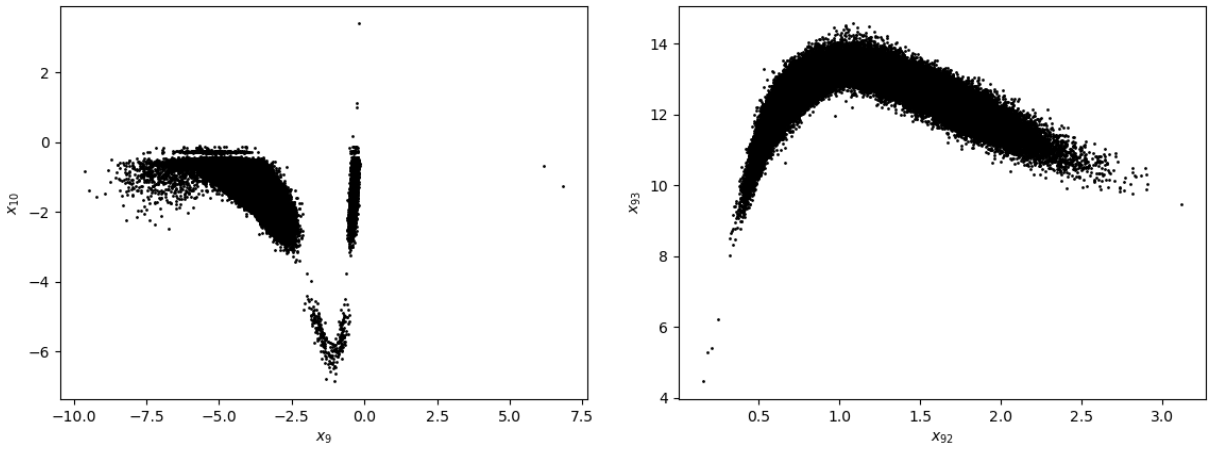

Figure 6: Scatter plots of the variates $x_{9:10}^{[i]}$ and $x_{92:93}^{[i]}$ from the mixture density network XMC simulation smoother, which is based on the nonlinear model in (13) and the simulated observations shown in Figure 4 (a).

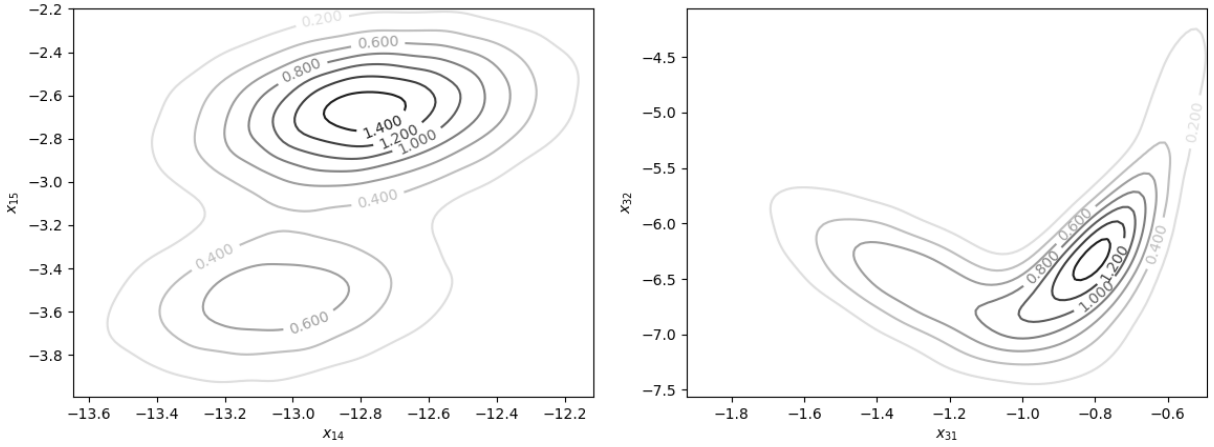

Figure 7: Contour plots of the kernel density estimates $\widehat{f}(x_{14:15}|y)$ and $\widehat{f}(x_{31:32}|y)$ for variate pairs from the mixture density network XMC simulation smoother, which is based on the nonlinear model in (13) and the simulated observations shown in Figure 4 (a).

4 Empirical application: Bitcoin volatility smoothing

As empirical application, we consider simulation smoothing for the volatility of Bitcoin log returns from December 20^{th}, 2020 to July 7^{th}, 2021 ($T = 200$), shown in Figure 8 (a). To specify the time-varying volatility, the following stochastic volatility (SV) model with stable errors is used (Lombardi & Calzolari, 2009; Vankov, Guindani, & Ensor, 2019),

$$\begin{aligned} y_t &= \exp(x_t/2)\varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{S}(\alpha, \beta), \\ x_{t+1} &= \mu_x + \phi_x(x_t - \mu_x) + \sigma_x \varepsilon_t^x, & \varepsilon_t^x &\sim \mathrm{N}(0, 1), \end{aligned} \tag{14}$$

with initialization $x_1 \sim \mathrm{N}\big(\mu_x, \sigma_x^2/(1-\phi_x^2)\big)$ and static parameters $\mu_x \in \mathbb{R}$, $|\phi_x| < 1$, and $\sigma_x > 0$. In addition, $\mathrm{S}(\alpha, \beta)$ denotes the first parameterization of the stable distribution as in Nolan (2009), with tail index $\alpha \in (0, 2]$ and asymmetry parameter $\beta \in [-1, 1]$. Apart

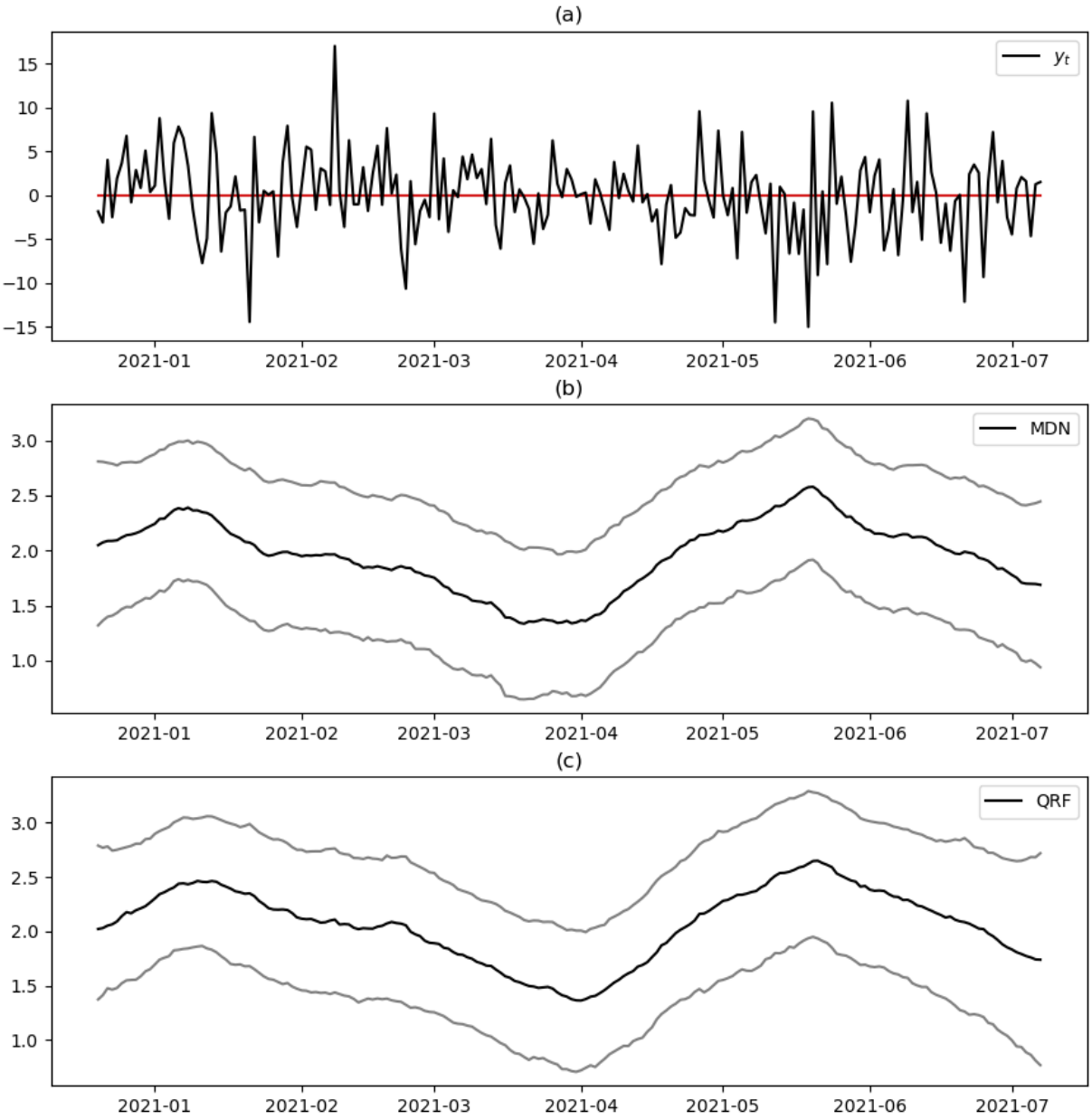

Figure 8: Bitcoin log returns y_t and marginal smoothing quantiles for cumulative probabilities $10\%, 50\%$, and 90% of the state from the SV model in (14) estimated by the XMC simulation smoother with $N = 10^6$ and $D = 10^4$: (a) log returns times 100; (b) quantiles for the mixture density network (MDN) version; (c) quantiles for the quantile regression forest (QRF) version.

from a few specific parameter choices (e.g., $\alpha = 2$, which yields the normal distribution), the density $p(\varepsilon_t^y) = p(y_t|x_t)$ is intractable, which precludes the direct use of standard Bayesian and SMC methods. The distribution is described by its characteristic function,

$$\mathbb{E}[\exp(iu\varepsilon_t^y)] = \begin{cases} \exp\left(-|u|\left[1 + i\beta\frac{2}{\pi}(\operatorname{sgn} u)\log|u|\right]\right) & \text{if } \alpha = 1, \\ \exp\left(-|u|^\alpha\left[1 - i\beta\tan(\frac{\pi\alpha}{2})(\operatorname{sgn} u)\right]\right) & \text{otherwise.} \end{cases} \tag{15}$$

Simulation is straightforward via the method of Chambers, Mallows, and Stuck (1976).

An important advantage of the XMC method for simulation smoothing is that it

remains applicable in such settings of limited tractability, as it requires only the ability to draw paths from the SSM. The MDN- and QRF-XMC simulation smoothers were applied to the SV model and Bitcoin data described above, with $N = 10^6$ and $D = 10^4$. The static parameters were set to the indirect inference estimates from Blasques, Koopman, and Moussa (2023b): $\alpha = 1.677$, $\beta = -0.061$, $\mu_x = 1.910$, $\phi_x = 0.994$, and $\sigma_x = 0.165$. The estimated $10\%, 50\%$, and 90% marginal smoothing quantiles are shown in Parts (b) and (c). The results for the two versions are seen to be very similar, which provides a validation of their accuracy. The quantiles are higher near the ends of the sample, which is consistent with the relatively calm middle region of the returns.

Simulation smoothing also allows for estimating nonlinear functions of the states conditional on the data. As illustration, Figure 9 shows the first 10 smoothed paths of the volatility, $\sigma_t = \exp(x_t/2)$, and their means based on all $D = 10^4$ paths. The smoothed volatilities are similar and higher near the ends of the sample, as for the state quantiles.

To illustrate the SS approach from Section 2.2, it was used to repeat the application corresponding to Figure 8 with $c_{\text{ss}} = 0$. The resulting state quantiles are shown in Figure 10, which are seen to be very similar to the regular estimates in Figure 8. Table 2 provides the average validation loss for the regular and SS approaches. For the MDN-XMC simulation smoother, the SS approach outperformed the regular version, which suggests some degree of overfitting of the training sample by the regular version. For the

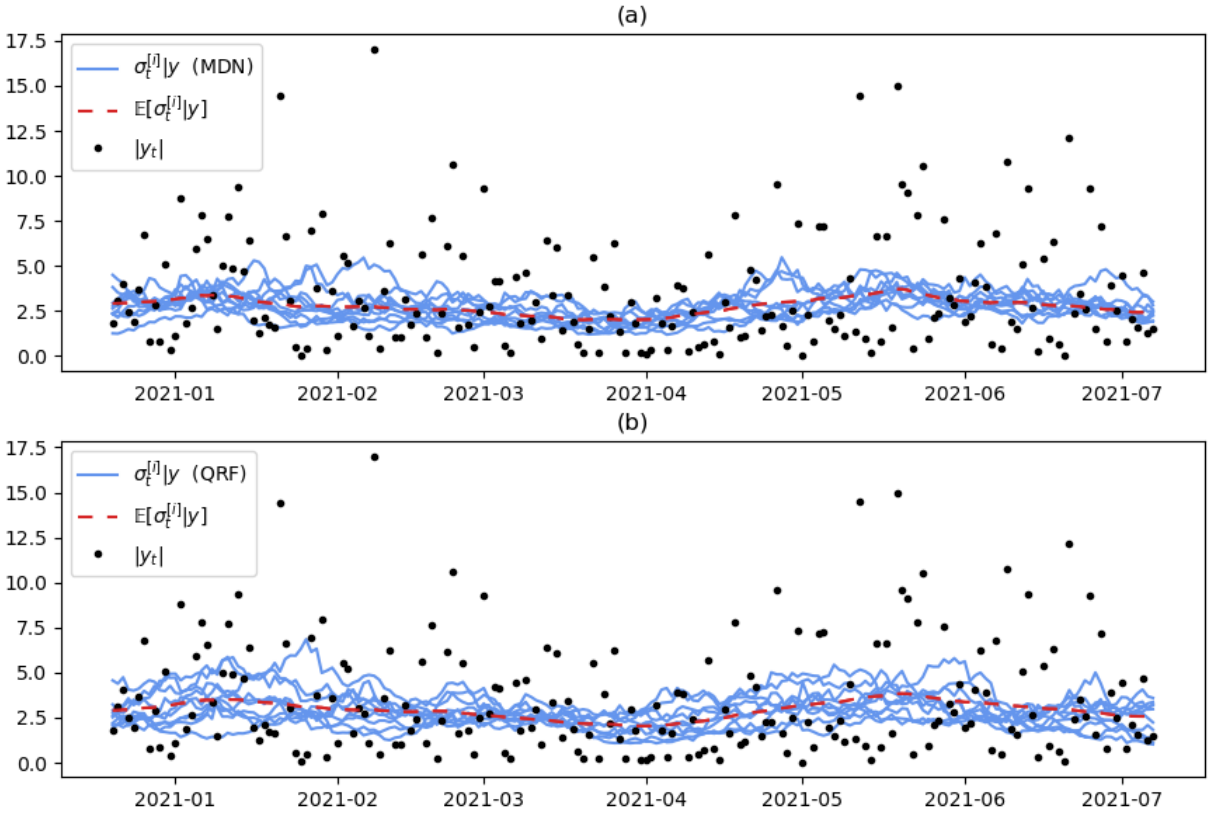

Figure 9: Paths of the volatility $\sigma_t = \exp(x_t/2)$ from the XMC simulation smoother based on the SV model in (14) and the Bitcoin data: (a) mean volatilities and first 10 out of $D = 10^4$ paths used to compute them for the mixture density network (MDN) version of the XMC simulation smoother; (b) analogous quantities for the quantile regression forest (QRF) version.

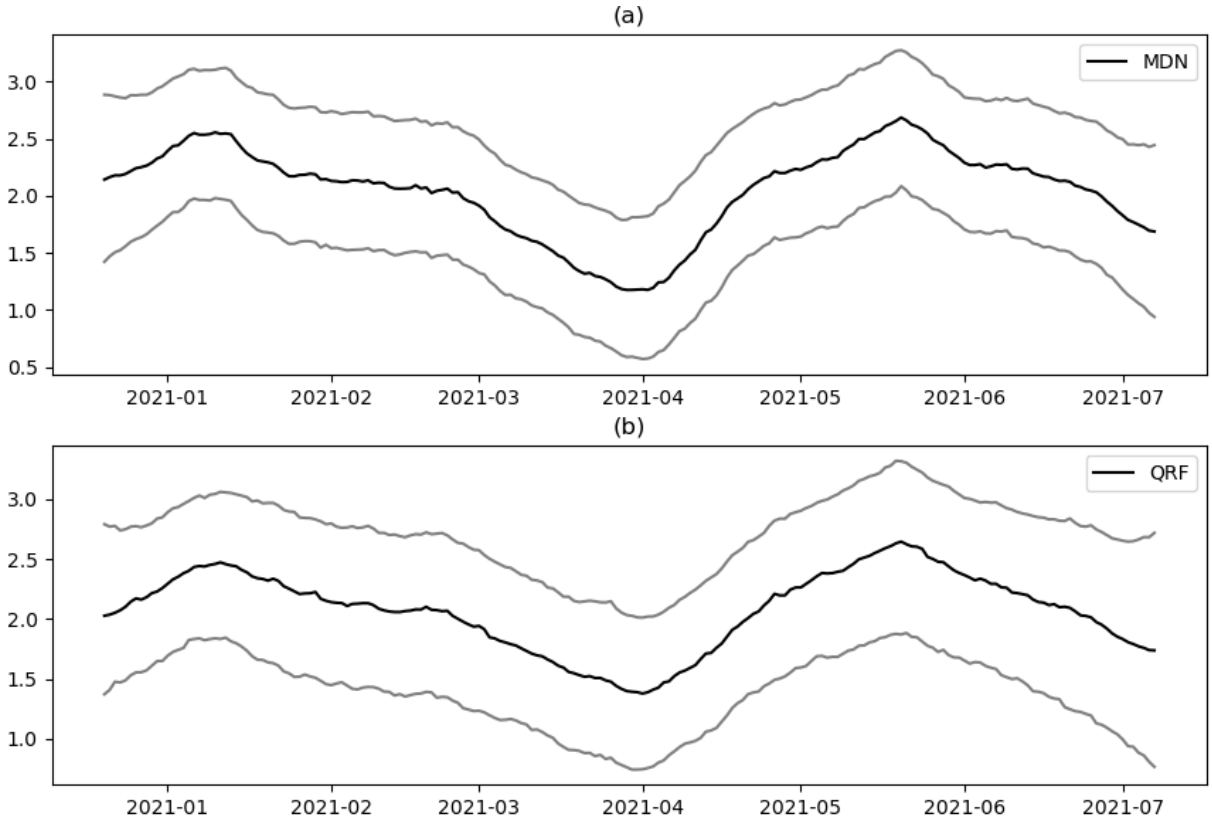

Figure 10: Marginal smoothing quantiles for cumulative probabilities $10\%, 50\%$, and 90% of the state from the SV model in (14) estimated by the steady state XMC simulation smoother with $c_{\text{ss}} = 0, N = 10^6$, and $D = 10^4$ applied to the Bitcoin data: (a) quantiles for the mixture density network (MDN) version; (b) quantiles for the quantile regression forest (QRF) version.

QRF version, the SS approach has effectively no impact on the performance. The SS was reached at the first opportunity ($t_{\text{ss}} = 199$) for the MDN version and shortly thereafter ($t_{\text{ss}} = 191$) for the QRF version. This resulted in savings of 93.5% and 85.5% of the $T = 200$ maximum possible estimations, respectively.

5 Conclusion

This paper has introduced a novel simulation smoothing method for state space models. The method can be used to compute smoothed estimates of the states and nonlinear functions of the states, and it allows for visualization of the joint smoothing distribution. The

Table 2: Estimation results for the regular and steady state (SS) approach to XMC simulation smoothing for the stable SV model in (14).

		XMC algorithm			
Method	Reported loss	Regular	SS	t_{ss}	Savings
MDN	Negative average log likelihood	-0.419	-0.423	199	93.5%
QRF	Root mean squared error	0.026	0.026	191	85.5%

simulation smoother is based on the extremum Monte Carlo method. It uses simulated data from the model to estimate the conditional density functions from the backward decomposition of the joint smoothing density. The approach is generally applicable and deals naturally with missing data, as well as measurements that become available at mixed frequencies. The method is illustrated via examples with missing data, multimodal distributions, and intractable model densities. The flexibility and computational efficiency of the approach is demonstrated in an empirical application to a time series of Bitcoin based on the stochastic volatility model with stable errors.

References

Amemiya, T. (1985). *Advanced Econometrics*. Harvard University Press.

Anderson, B., & Moore, J. B. (1979). Optimal filtering. *Prentice-Hall*.

Ankargren, S., & Jonéus, P. (2021). Simulation smoothing for nowcasting with large mixed-frequency vars. *Econometrics and Statistics*, *19*, 97–113.

Barndorff-Nielsen, O. E., & Shephard, N. (2001). *Normal modified stable processes*. Citeseer.

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning* (pp. 115–123).

Bishop, C. M. (1994). Mixture density networks.

Blasques, F., Koopman, S. J., & Moussa, K. (2023a). Extremum Monte Carlo filters: Real-time signal extraction via simulation and regression. *Discussion Paper Tinbergen Institute TI 2023-016/III*. Retrieved from `https://papers.tinbergen.nl/23016.pdf`

Blasques, F., Koopman, S. J., & Moussa, K. (2023b). Stochastic Volatility with Stable Errors: Estimation, Filtering and Forecasting. *Financial Econometrics Conference Lancaster University*. Retrieved from `http://wp.lancs.ac.uk/finec2023/files/2023/01/FEC-2023-093-Siem-Jan-Koopman.pdf`

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Carter, C. K., & Kohn, R. (1994). On gibbs sampling for state space models. *Biometrika*, *81*(3), 541–553.

Chambers, J. M., Mallows, C. L., & Stuck, B. (1976). A method for simulating stable random variables. *Journal of the American Statistical Association*, *71*(354), 340–344.

Chopin, N., & Papaspiliopoulos, O. (2020). *An introduction to sequential monte carlo*. Springer.

Creal, D. D. (2008). Analysis of filtering and smoothing algorithms for lévy-driven stochastic volatility models. *Computational Statistics & Data Analysis*, *52*(6), 2863–2876.

De Jong, P., & Shephard, N. (1995). The simulation smoother for time series models. *Biometrika*, *82*(2), 339–350.

Doucet, A., De Freitas, N., & Gordon, N. J. (2001). *Sequential Monte Carlo methods in practice* (Vol. 1) (No. 2). Springer.

Durbin, J., & Koopman, S. J. (1997). Monte carlo maximum likelihood estimation for non-gaussian state space models. *Biometrika*, *84*(3), 669–684.

Durbin, J., & Koopman, S. J. (2002). A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, *89*(3), 603–616.

Durbin, J., & Koopman, S. J. (2012). *Time series analysis by state space methods*. Oxford University Press.

Fearnhead, P., Wyncoll, D., & Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika*, *97*(2), 447–464.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, *15*(2), 183–202.

Galilei, G. (1632). *Dialogo sopra i due massimi sistemi del mondo, tolemaico, e copernicano*.

Gauss, C. F. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 1317–1339.

Godsill, S. J., Doucet, A., & West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, *99*(465), 156–168.

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEEE Proceedings F (radar and signal processing)* (Vol. 140, pp. 107–113).

Hald, A. (1986). Galileo's statistical analysis of astronomical observations. *International Statistical Review/Revue Internationale de Statistique*, 211–220.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Hoogerheide, L. F., Van Dijk, H. K., & Van Oest, R. D. (2009). Simulation based bayesian econometric inference: principles and some recent computational advances. *Handbook of Computational Econometrics*, 215–280.

Jacquier, E., Polson, N. G., & Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, *12*, 371–389.

Kim, S., Shephard, N., & Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with arch models. *The Review of Economic Studies*, *65*(3), 361–393.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, *5*(1), 1–25.

Legendre, A. M. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes*. Courcier.

Lin, C.-C., & Eisner, J. (2018). Neural particle smoothing for sampling from conditional sequence models. *arXiv preprint arXiv:1804.10747*.

Lombardi, M. J., & Calzolari, G. (2009). Indirect estimation of α-stable stochastic volatility models. *Computational Statistics & Data Analysis*, *53*(6), 2298–2308.

Mardia, K. V., Jupp, P. E., & Mardia, K. (2000). *Directional statistics* (Vol. 2). Wiley Online Library.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(6).

Moussa, K. (2024). *Signal Extraction by the Extremum Monte Carlo Method* (PhD dissertation, Vrije Universiteit Amsterdam and Tinbergen Institute, the Netherlands).

Retrieved from https://research.vu.nl/ws/portalfiles/portal/301330660/thesiskmoussawithcover+-+65d85ac1813ea.pdf

Nolan, J. P. (2009). Univariate stable distributions. *Stable Distributions: Models for Heavy Tailed Data*, *22*(1), 79–86.

Paige, B., & Wood, F. (2016). Inference networks for sequential monte carlo in graphical models. In *International conference on machine learning* (pp. 3040–3049).

Rubin, D. B. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. *Journal of the American Statistical Association*, *82*(398), 543–546.

Shephard, N., & Pitt, M. K. (1997). Likelihood analysis of non-gaussian measurement time series. *Biometrika*, *84*(3), 653–667.

Smith, A. F., & Gelfand, A. E. (1992). Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, *46*(2), 84–88.

Stuhlmüller, A., Taylor, J., & Goodman, N. (2013). Learning stochastic inverses. *Advances in neural information processing systems*, *26*.

Vankov, E. R., Guindani, M., & Ensor, K. B. (2019). Filtering and estimation for a class of stochastic volatility models with intractable likelihoods. *Bayesian Analysis*, *14*(1), 29–52.

Whittaker, E. T. (1923). On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, *41*, 63–75.

Appendix A Estimators of conditional distributions

A.1 Quantile regression forest

To discuss the quantile regression forest (QRF; Meinshausen, 2006), we start by considering the random forest (RF; Breiman, 2001), which serves as its key component. The RF is defined as an average of a large number of decorrelated regression trees that are grown using bootstrapped samples of the data. The idea is that regression trees are characterized by a low bias and high variance, and because the trees are identically distributed their average retains this low bias while benefiting from a reduced variance. A distinctive step in the RF procedure is that when growing a tree, each split decision in a terminal node is made using a subset of randomly selected covariates to reduce the correlation between the trees. Given a sample $(X^{(i)}, Y^{(i)}),\ i = 1, \ldots, N$ of random variables X and Y, the RF predictions of X can be represented directly in terms of the data by

$$\sum_{i=1}^{N} w_i(y) X^{(i)} \approx \mathbb{E}[X|Y = y], \qquad \sum_{i=1}^{N} w_i(y) = 1,\ w_i(y) \geq 0.$$

The weights are defined as an average over the weights of K regression trees,

$$w_i(y) = \frac{1}{K} \sum_{k=1}^{K} w_i^k(y), \tag{16}$$

where each tree predicts the dependent variable X by taking the average over the corresponding variates in the leaf to which y is assigned. This may be represented by

$$w_i^k(y) = \frac{1_{\{i \in \mathbb{L}_k(y)\}}}{|\mathbb{L}_k(y)|},$$

with $\mathbb{L}_k(y)$ the set of indices j corresponding to the values $Y^{(j)}$ in the leaf to which y is assigned and $|\mathbb{L}_k(y)|$ its number of elements.

The QRF method exploits the RF to perform quantile regression. Noting that for random variables X and Y the conditional CDF is defined by

$$P(X \leq x|Y = y) = \mathbb{E}[1_{\{X \leq x\}}|Y = y],$$

the QRF approximates the right-hand side by

$$\sum_{i=1}^{N} w_i(y)\ 1_{\{X^{(i)} \leq x\}}, \tag{17}$$

with the weights as defined in (16). The quantile estimates can then be computed by inverting the above CDF estimate. This approach requires only a single RF to be fit for estimating any number of quantiles, and it ensures monotonicity in the cumulative probabilities. Convergence of the QRF method is discussed in Section 4 of Meinshausen (2006). The CDF estimator in (17) implies the following estimator of the PMF:

$$\widehat{f}^N(x|Y = y) = \sum_{i=1}^{N} w_i(y)\ 1_{\{X^{(i)} = x\}}.$$

Drawing from the above PMF is a straightforward exercise in resampling the variates $X^{(i)}$ with replacement using probabilities $w_i(y)$.

A.2 Mixture density network

To discuss the mixture density network (MDN; Bishop, 1994), suppose our aim is to estimate the conditional density $p(X|Y)$ for random vectors $X \in \mathbb{R}^{N_X}$ and $Y \in \mathbb{R}^{N_Y}$. Consider the mixture of normal densities,

$$f(X) = \sum_{j=1}^{N_c} w_j \, p_{\mathrm{N}}(X; \mu_j, \sigma_j^2 I_{N_X}), \qquad \sum_{j=1}^{N_c} w_j = 1, \; w_j \geq 0,$$

with N_c components and p_{N} the multivariate normal density. In the MDN, the parameters $w_j = w_j(y)$, $\mu_j = \mu_j(y)$, and $\sigma_j = \sigma_j(y)$ are defined as the outputs of a neural network (e.g., Hastie, Tibshirani, & Friedman, 2009, Ch. 11) with inputs $y \in \mathbb{R}^{N_y}$. This defines a conditional density, which can be estimated via maximum likelihood given a sample $(X^{(i)}, \; Y^{(i)}), \; i = 1, \ldots, N$. The estimator is

$$\widehat{f}^N \in \operatorname*{arg\,max}_{f \in \mathbb{F}_N} \frac{1}{N} \sum_{i=1}^{N} \log f\left(X^{(i)} | Y^{(i)}\right),$$

where $\mathbb{F}_N$ is the set of MDNs considered. The universal approximation properties of neural networks and mixtures of Gaussian densities ensure that the MDN estimator is generally applicable (Bishop, 1994).

Appendix B *N*-convergence

In this section we consider convergence of the XMC simulation smoother as the number of simulated paths, N, diverges to infinity. It will be shown that under appropriate conditions, the XMC estimator of the joint smoothing density from Algorithm 2,

$$\widehat{f}^N(x|y) = \prod_{t=1}^{T} \widehat{f}_t^N(x_t | \widetilde{Y}_t^N), \tag{18}$$

converges to the target smoothing density, $p(x|y)$, as $N \to \infty$. Here, the notation $\widetilde{Y}_t^N$ makes explicit that we allow the covariate sets to depend on N, which covers the out-of-sample optimization procedure from Section 2.1.

Let $\mathcal{X} = \{ x \,|\, p(x) > 0 \}$ denote the set of all realizable paths of the states (Frühwirth-Schnatter, 1994), with $\mathcal{Y} = \{ y \,|\, p(y) > 0 \}$ defined by analogy for the paths of the observations. We then assume that the joint smoothing density $p(x|y)$ is defined for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, such that

$$p(x|y) = \prod_{t=1}^{T} p(x_t | Y_t), \qquad \int_{x \in \mathcal{X}} p(x|y) dx = 1, \qquad p(x|y) \geq 0, \tag{19}$$

with Y_t the conditioning set from the backward decomposition in (5) at time t. The above decomposition corresponds to the standard assumptions that the states follow a Markov process and the noise terms are independent. If needed, these assumptions can be dropped by considering the more general backward decompositions in (3) or (4).

We shall say that the XMC estimator in (18) converges to the joint smoothing density if for all $y \in \mathcal{Y}$,

$$\sup_{x \in \mathcal{X}} \left| \widehat{f}^N(x|y) - p(x|y) \right| \xrightarrow{P} 0 \qquad \text{as} \qquad N \to \infty. \tag{20}$$

Here, the convergence in probability pertains to the random nature of the training and validation samples used in Algorithm 2. Note that the above convergence is pointwise over the realizable paths of the observations y (i.e., the observed data) and uniform over the realizable paths of the states x. While the support of both x and y may be (and typically is) non-compact, uniform convergence is reasonable only for the state paths because $p(x|y)$ is integrable with respect to x, hence it must converge to zero when letting the states go to $\pm\infty$.

We have the following convergence result for the XMC simulation smoother.

Theorem 1 (Simulation smoother N-convergence)**.** *Suppose the following holds:*

1.1 The joint smoothing density $p(x|y)$ is defined as in (19)*, and its components $p(x_t|Y_t)$ are bounded with respect to x_t, such that*[2]

$$\sup_t \sup_{x \in \mathcal{X}} p(x_t|Y_t) < \infty.$$

1.2 There exists some rate $r_N > 0$ that diverges as $N \to \infty$, such that for all $y \in \mathcal{Y}$ and $t = 1, \ldots, T$,

$$r_N \sup_{x \in \mathcal{X}} \left| \widehat{f}_t^N\left(x_t|Y_t\right) - p(x_t|Y_t) \right| = O_P(1).$$

1.3 The components from the XMC estimator of the joint smoothing density in (18) *are bounded, such that for every $y \in \mathcal{Y}$ and covariate set $\widetilde{Y}_t^N$,*

$$\sup_t \sup_{x \in \mathcal{X}} \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) < \infty.$$

A.1.4 For $t = 1, \ldots, T$ the regularized covariate set converges in probability to the conditioning set,

$$\lim_{N \to \infty} P(\widetilde{Y}_t^N = Y_t) = 1.$$

Then the XMC estimator in (18) *converges in probability to the joint smoothing density, $p(x|y)$, such that the condition in* (20) *is satisfied for all $y \in \mathcal{Y}$. Furthermore, convergence occurs at rate r_N,*

$$r_N \sup_{x \in \mathcal{X}} \left| \widehat{f}^N(x|y) - p(x|y) \right| = O_P(1).$$

[2]Note that because x_t is an element of the path x, we can consider the supremum $\sup_{x \in \mathcal{X}} g(x_t)$ for any function $g(x_t)$.

Proof. By Assumptions 1.1 and 1.3, there exists some upper bound $U > 0$ for the density components and their estimators, such that

$$\sup_t \sup_{x \in \mathcal{X}} \max \left\{ p(x_t|Y_t),\ \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) \right\} < U. \tag{21}$$

In the following, we assume without loss of generality that $U \leq 1$. We then have for all $y \in \mathcal{Y}$ that

$$\begin{aligned}
&\sup_{x \in \mathcal{X}} \left| \widehat{f}^N(x|y) - p(x|y) \right| = \sup_{x \in \mathcal{X}} \left| \prod_{t=1}^T \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) - \prod_{t=1}^T p(x_t|Y_t) \right| \\
\leq &\sup_{x \in \mathcal{X}} \sum_{t=1}^T \left| \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) - p(x_t|Y_t) \right| \leq \sum_{t=1}^T \sup_{x \in \mathcal{X}} \left| \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) - p(x_t|Y_t) \right| \\
\leq &\sum_{t=1}^T \sup_{x \in \mathcal{X}} \left(\left| \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) - \widehat{f}_t^N(x_t|Y_t) \right| + \left| \widehat{f}_t^N(x_t|Y_t) - p(x_t|Y_t) \right| \right) \\
\leq &\sum_{t=1}^T \sup_{x \in \mathcal{X}} \left| \widehat{f}_t^N(x_t|\widetilde{Y}_t^N) - \widehat{f}_t^N(x_t|Y_t) \right| + \sum_{t=1}^T \sup_{x \in \mathcal{X}} \left| \widehat{f}_t^N(x_t|Y_t) - p(x_t|Y_t) \right| \\
= &o_P(r_N^{-1}) + O_P(r_N^{-1}) = O_P(r_N^{-1}),
\end{aligned}$$

where the first equality follows by definition from (18) and Assumption 1.1, while the first inequality follows from the identity

$$\left| \prod_{t=1}^T a_t - \prod_{t=1}^T b_t \right| \leq \sum_{t=1}^T |a_t - b_t|,$$

which holds for any two sequences of complex numbers $\{a_t\}_{t=1}^T$ and $\{b_t\}_{t=1}^T$ with $|a_t| < 1$ and $|b_t| < 1$ for $t = 1, \ldots, T$. The above identity can be used because it was assumed that $U \leq 1$ in (21). This assumption is innocuous because if $U > 1$ we can write

$$\sup_{x \in \mathcal{X}} \left| \widehat{f}^N(x|y) - p(x|y) \right| = U^T \sup_{x \in \mathcal{X}} \left| U^{-T} \left(\widehat{f}^N(x|y) - p(x|y) \right) \right|$$

and proceed as above. The second and fourth inequalities hold because the sum over the supremum of the terms separately cannot be lower than the supremum of the sum, and the third inequality follows from the triangle inequality. In the final upper bound, the terms of the second sum are $O_P(r_N^{-1})$ by Assumption 1.2. Furthermore, the terms of the first sum are $o_P(r_N^{-1})$, which we shall now demonstrate. For conciseness, let

$$\Delta(\widetilde{Y}_t) = \sup_{x \in \mathcal{X}} \left| \widehat{f}_t^N(x_t|\widetilde{Y}_t) - \widehat{f}_t^N(x_t|Y_t) \right|,$$

and let the set of all feasible covariate sets for a given conditioning set Y_t be the power set $\mathcal{P}(Y_t)$ (i.e., the set of all subsets of Y_t). Then,

$$\Delta(\widetilde{Y}_t^N) \leq \sup_{\widetilde{Y}_t \in \mathcal{P}(Y_t)} \Delta(\widetilde{Y}_t) \cdot 1_{\{\widetilde{Y}_t^N \neq Y_t\}},$$

since $\widetilde{Y}_t^N \in \mathcal{P}(Y_t)$ and the left-hand side evaluates to zero when $\widetilde{Y}_t^N = Y_t$. It follows that for any $\epsilon > 0$,

$$\begin{aligned} P(r_N \cdot \Delta(\widetilde{Y}_t^N) > \epsilon) &\leq P\left(r_N \cdot \sup_{\widetilde{Y}_t \in \mathcal{P}(Y_t)} \Delta(\widetilde{Y}_t) \cdot 1_{\{\widetilde{Y}_t^N \neq Y_t\}} > \epsilon\right) \\ &\leq P\left(r_N \cdot \sup_{\widetilde{Y}_t \in \mathcal{P}(Y_t)} \Delta(\widetilde{Y}_t) \cdot 1_{\{\widetilde{Y}_t^N \neq Y_t\}} > 0\right) \\ &\leq P(\widetilde{Y}_t^N \neq Y_t) \to 0 \qquad \text{as} \qquad N \to \infty, \end{aligned}$$

where the convergence step follows from Assumption A.1.4. We therefore have that $\Delta(\widetilde{Y}_t^N) = o_P(r_N^{-1})$, and since the above applies to every time t, the XMC estimator in (18) converges in probability to the joint smoothing density at rate r_N.

□

As mentioned above, by modifying the covariate sets, the first part of Assumption 1.1 can be weakened to allow for non-Markov state processes and dependence between the noise terms. The boundedness assumption holds when the densities $p(x)$ and $p(y|x)$ are bounded in the states path x. In cases where this assumption is violated, we can typically approximate the density arbitrarily well by a bounded density.[3]

Assumption 1.2 states that when the conditioning set is used as covariate set ($\widetilde{Y}_t^N = Y_t$), the XMC function estimators must converge in probability to the true densities at some rate r_N. The rate r_N is allowed to differ from the usual parametric rate $\sqrt{N}$, since the XMC simulation smoother will often use (semi-)nonparametric estimators.

Assumption 1.3 holds for the MDN estimator under the mild restriction that the component densities are bounded, which is ensured in the Gaussian mixture case when the variances of the component densities are bounded below by some arbitrary small positive value. The assumption also holds for conditional density estimators based on KDEs with bounded kernels. Furthermore, it holds in the discrete case (e.g., the QRF estimator) where $\widehat{f}_t^N$ is a PMF, which is bounded above by one by definition.

Assumption 1.4 ensures that the use of a regularized covariate set does not impact the convergence of the simulation smoother or its rate. With the out-of-sample optimization procedure from Section 2.1, this is guaranteed when none of the observations can be omitted from the full covariate set, $\widetilde{Y}_t^N = Y_t$, without increasing the expected loss for $t = 1, \ldots, T$. The latter generally holds when the state has autoregressive dynamics, which applies to most SSMs used in practice. On the other hand, if there exists a proper subset $\widetilde{Y}_t \subset Y_t$ for which the minimum expected loss is attained,[4] then Assumption 1.4 is of course no longer necessary for convergence.

[3]For example, the Gamma density, $p(z) = \frac{1}{\Gamma(k)\sigma^k} z^{k-1} \exp(-z/\sigma)$, with support $z \in (0, \infty)$ is unbounded for $k < 1$ because $\lim_{z \downarrow 0} p(z) = \infty$, but this issue can be circumvented by restricting the support to (ϵ, ∞) for an arbitrarily small $\epsilon > 0$.

[4]A trivial example is a SSM with $y_t = x_t = x_1 \ \forall \ t$, such that any single observation y_t contains all the information in Y_t. Another example is when the state follows a finite-order moving average process.