TI 2023-016/III
Tinbergen Institute Discussion Paper

Extremum Monte Carlo Filters: Signal Extraction via Simulation and Regression

Revision: May 2025

Karim Moussa[1]
Francisco Blasques[2]
Siem Jan Koopman[3]

[1] Vrije Universiteit Amsterdam, Tinbergen Institute

[2] Vrije Universiteit Amsterdam, Tinbergen Institute

[3] Vrije Universiteit Amsterdam, Tinbergen Institute

Extremum Monte Carlo Filters:
Signal Extraction via Simulation and Regression

Karim Moussa, Francisco Blasques, Siem Jan Koopman[*]

Vrije Universiteit Amsterdam and Tinbergen Institute, the Netherlands

First version: November 18, 2021
This version: May 1, 2025

Abstract

We introduce a novel simulation-based method for signal extraction in a general class of state space models. It can be used to estimate time-varying conditional means, modes, and quantiles, and to predict latent variables or forecast observations. The method consists of generating artificial data sets from the model and estimating the quantities of interest via extremum estimation. The approach is broadly applicable and its implementation is straightforward. The method is suited for signal extraction in cases of long time series, missing data, or high-dimensionality. Furthermore, we demonstrate its use in real-time filtering, where most of the computations can be performed in advance, and in fixed-interval smoothing. Conditions for the stability and convergence of the filtering method are discussed, and its key properties are illustrated by various applications, including nonlinear and high-dimensional models.

Keywords: Fixed-interval smoothing, Latent variables, Least squares Monte Carlo, Multivariate stochastic volatility, Real-time filtering, State space models.

[*] Corresponding author. E-mail: `s.j.koopman@vu.nl`.

1 Introduction

State space models (SSMs) decompose observed time series into two unobserved parts: the states (or signal), which are the true objects of interest, and the noise, which complicates the extraction of the signal from the data. The state space modeling approach has become pervasive in both the scientific and industry domains, with applications in fields varying from financial econometrics and forecasting to robotics (Doucet, De Freitas, and Gordon 2001; Durbin and Koopman 2012; Chopin and Papaspiliopoulos 2020).

Central to many of these applications is the task of signal extraction, which involves estimating the unobserved signals based on noisy measurements. This is widely recognized as a challenging problem, for which numerous approaches have been developed. Early influential work focused on linear prediction, where Kolmogorov (1941) and Wiener (1949) considered its application to stationary processes. The seminal work of Kalman (1960) enabled the treatment of non-stationarity within the framework of linear SSMs.

Real-world applications often involve nonlinearity and non-Gaussian distributions, in which case linear prediction methods are typically inadequate. As a result, much subsequent work has focused on nonlinear signal extraction. Notable examples include the extended Kalman filter (e.g., Anderson & Moore, 1979, Ch. 8.2), the unscented Kalman filter (Julier & Uhlmann, 1997), Gaussian-sum filters (Sorenson & Alspach, 1971), and methods based on numerical integration (e.g., Kitagawa, 1987). In addition, many modern approaches to signal extraction rely on simulation, taking advantage of recent increases in computing power. These include Markov chain Monte Carlo methods (Andrieu, Doucet, & Holenstein, 2010), importance sampling (Durbin & Koopman, 2012, Ch. 11), and particle filtering (e.g., Gordon, Salmond, and Smith 1993; Pitt and Shephard 1999; Creal 2012).

This paper proposes a novel simulation-based signal extraction method for a general class of SSMs. The method involves generating artificial samples of data from the model and estimating the signals via extremum estimation (e.g., Amemiya, 1985), which includes least squares and maximum likelihood estimation as special cases. Hence, we refer to this procedure as *extremum Monte Carlo* (XMC). We use it to define a corresponding class of filtering and smoothing methods that can be used generally, with the minimum requirement of being able to simulate from the model. The implementation of XMC is straightforward, providing an accessible approach to signal extraction for nonlinear and non-Gaussian models. Given that most of the computations can be performed in advance, the filtering method is well-suited for real-time applications, such as recommender systems in e-commerce (Schafer, Konstan, & Riedl, 1999) and algorithmic trading in finance (Kolm & Maclin, 2010). In addition, the method can be employed for forecasting and fixed-interval smoothing.

The remainder of this paper is structured as follows. Section 2 introduces the SSM and provides the motivation for this work. Section 3 presents the XMC method and its extensions. Section 4 provides a stability and convergence analysis. Section 5 illustrates the key properties of the method through various applications. Section 6 provides discussion and Section 7 concludes. The appendix contains proofs and other supplementary material.

2 State Space Model and Motivation

2.1 State Space Model

Let $x_t \in \mathbb{R}^{N_x}$ denote the state vector at time t, and let $y_t \in \mathbb{R}^{N_y}$ be the corresponding vector of measurements (or observations) for some $N_x, N_y \in \mathbb{N}$, with the related noise vectors denoted by ε_t^x and ε_t^y, respectively. We consider the SSM as given by

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

for $t = 1, \dots, T$, where $T \in \mathbb{N}$ is the length of the time series, m_t and s_t are the measurement and state transition functions, respectively, and $p(\cdot)$ denotes the probability density of the corresponding variable, which may be non-Gaussian. We assume that the SSM allows for simulating paths of the states, $x_{1:T} = (x_1, \dots, x_T)$, and the observations, $y_{1:T}$. The functions

m_t and s_t in (1) can be nonlinear, and they may depend on exogenous variables, lags of the states and observations, and on a vector of static parameters, θ, which is assumed to be fixed and given. The probability density functions p may also depend on θ.

Signal extraction is typically performed using the conditional expectation of the states,

$$\mathbb{E}\left[x_t | \mathcal{F}_t\right], \tag{2}$$

for $t = 1, \dots, T$, where $\mathcal{F}_t$ represents the information available for predicting the state at time t. Common choices for the information set include $\mathcal{F}_t = y_{1:t}$ for filtering, $\mathcal{F}_t = y_{1:t-k}$ with $k \in \mathbb{N}$ for k-period forecasting, and $\mathcal{F}_t = y_{1:T}$ for fixed-interval smoothing.

For linear Gaussian SSMs, the conditional expectations in (2) can be computed recursively by the Kalman filter. A simple example is the univariate local level model given by

$$\begin{aligned} y_t &= x_t + \varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{N}(0, \sigma_y^2), \\ x_{t+1} &= x_t + \varepsilon_t^x, & \varepsilon_t^x &\sim \mathrm{N}(0, \sigma_x^2), \end{aligned} \tag{3}$$

with $x_1 \sim \mathrm{N}(\mu_1, \sigma_1^2)$ for some $\mu_1 \in \mathbb{R}$ and $\sigma_1, \sigma_x, \sigma_y > 0$, and the scalar noise terms ε_t^x and ε_t^y are assumed to be normally distributed, mutually and serially independent, and independent from x_1. The local level model is a special case of the SSM in (1) with $m_t(x_t, \varepsilon_t^y) = x_t + \varepsilon_t^y$ and $s_t(x_t, \varepsilon_t^x) = x_t + \varepsilon_t^x$, joint density $p_{\mathrm{G}}(\varepsilon_t^x, \varepsilon_t^y) = p_{\mathrm{G}}(\varepsilon_t^x) \cdot p_{\mathrm{G}}(\varepsilon_t^y)$, where p_{G} denotes a Gaussian density, and $\theta = (\mu_1, \sigma_1, \sigma_x, \sigma_y)'$.

We consider an application of the local level model to annual flow volume measurements of the Nile River, ranging from 1871 to 1970 (Durbin & Koopman, 2012, Ch. 2). We set the static parameters σ_x and σ_y to their maximum likelihood estimates, $\sigma_x = 38.329$ and $\sigma_y = 122.877$, and set $\mu_1 = 0$ and $\sigma_1^2 = 10^7$ for approximate diffuse initialization. The filtering means $\mathbb{E}[x_t | y_{1:t}]$ are computed exactly using the Kalman filter for $t = 1, \dots, T$. The filtered states presented in Figure 1 (a) provide an estimate of the underlying trend based on the noisy measurements.

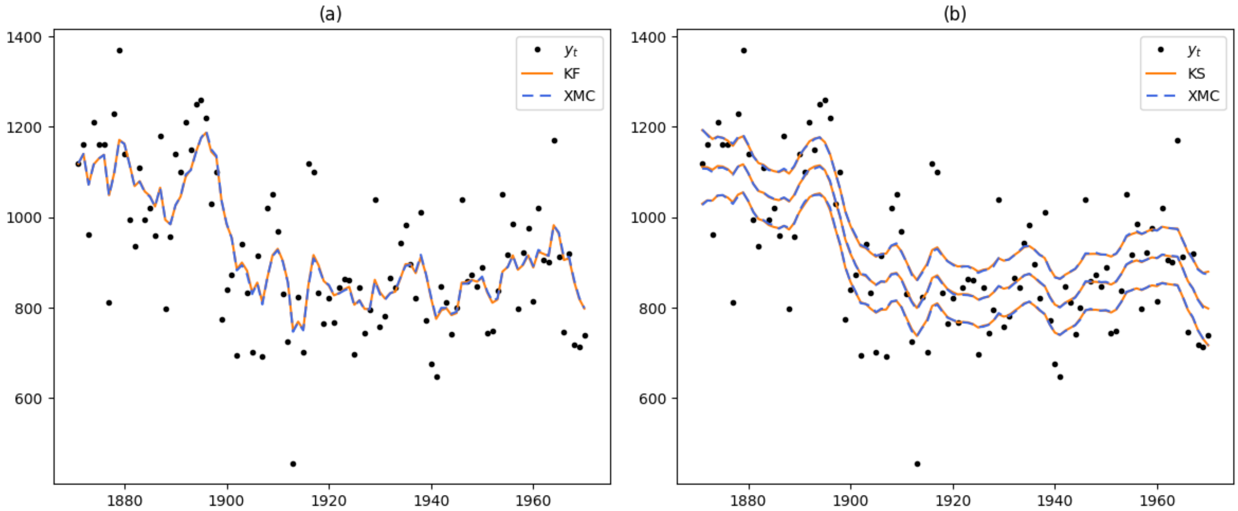


Figure 1: Analysis of the annual flow volume measurements y_t of the Nile river (discharge at Aswan in $10^8 m^3$) from 1871 to 1970 based on the local level model in (3): (a) signals extracted via $\mathbb{E}[x_t|y_{1:t}]$ by the Kalman filter (KF) and the linear extremum Monte Carlo (XMC) filter with $N = 5 \cdot 10^4$ paths and steady state reached at $t = 19$; (b) smoothing quantiles corresponding to $p(x_t|y_{1:T})$ for cumulative probabilities $10\%, 50\%$, and 90% by the Kalman smoother (KS) and the linear XMC smoother. The data are taken from Cobb (1978).

In practice, the linear Gaussian assumption is often violated; see Creal (2012) and Durbin and Koopman (2012) for multiple examples in economics and finance. For nonlinear and non-Gaussian SSMs, estimating the conditional expectation in (2) is a challenging task. This is typically addressed using simulation-based methods, such as particle filters, which have been successfully applied in a wide range of applications (e.g., Doucet et al., 2001). However, signal extraction remains difficult in scenarios where, for example, the model is high-dimensional or non-Markovian, the measurement or state transition densities are unavailable, or the conditional expectations must be evaluated sequentially in real time.

2.2 Extremum Monte Carlo: Background and Motivation

The XMC method finds its origins in the least squares Monte Carlo method of Longstaff and Schwartz (2001). This method was developed for pricing American options in financial trading. A crucial step in this pricing algorithm involves approximating a conditional expectation function, $\mathbb{E}[X|Y]$, using draws of the random variables X and Y,

$$X^{(i)},\ Y^{(i)}, \qquad i = 1, \dots, N.$$

The variates are then used as data in the following least squares regression,

$$\widehat{f}^N \in \operatorname*{arg\,min}_{f \in \mathbb{F}_N} \frac{1}{N} \sum_{i=1}^{N} L\Big(X^{(i)} - f\big(Y^{(i)}\big)\Big),$$

with $L(u) = u^2$ the squared error loss, $f(\cdot)$ a prediction function, and $\mathbb{F}_N$ a suitable function space. Finally, the function estimate is used to predict X for any Y value of interest via

$$\widehat{f}^N(y) \approx \mathbb{E}[X|Y = y].$$

In its simplest form, the XMC method involves repeatedly applying the above technique to perform signal extraction. Specifically, we set $X = x_t$ and $Y = \mathcal{C}_t \subseteq \mathcal{F}_t$ for $t = 1, \ldots, T$, where the covariates $\mathcal{C}_t$ are an appropriate subset of the information set at time t. In essence, we first use the SSM in (1) to simulate paths of the states and observations, and then regress the states onto subsets of the observations. The estimated regression functions are then evaluated with the actual data to predict the unobserved states.

The approach can be extended in various ways. For example, substantial computational savings can be achieved by reusing the function estimates for prediction at other time points. For an illustration, we return to the local level model example. Since the Kalman filter is linear in the observations, we can attempt to mimic this filter by applying the XMC method with linear regression to minimize the squared error loss for a sample of N simulated paths. Figure 1 (a) shows the filtered states based on the resulting linear XMC filter with $N = 5 \cdot 10^4$. The function estimate at $t = 19$ (the year 1889) was reused to filter the subsequent states. The filtered states visually match those of the Kalman filter.

By changing the loss function, the XMC method can be used to estimate other aspects of $p(x_t|\mathcal{F}_t)$. Notable examples include the tilted absolute error loss, $L_\tau(u) = u(\tau - 1_{\{u<0\}})$, with prediction error $u = X - f(Y)$, to estimate the conditional τ-quantile for $\tau \in (0, 1)$, and the all-or-nothing loss, $L_\delta(u) = 1_{\{|u| \geq \delta\}}$, with tolerance level $\delta > 0$, for approximating the conditional mode as $\delta \to 0$. To illustrate this, Figure 1 (b) shows the smoothing quantiles corresponding to $p(x_t|y_{1:T})$ for cumulative probabilities $10\%, 50\%$, and 90% based on the linear XMC smoother for quantile regression. The estimates match the exact quantiles obtained using the Kalman smoother (Durbin & Koopman, 2012, Ch. 4.4).

The key requirement for the XMC method is that the SSM in (1) can be used to simulate paths of the states and the observations, a condition that is satisfied in most applications. By employing suitable regression methods, the approach can be used for signal extraction with nonlinear and non-Gaussian SSMs.

3 The Extremum Monte Carlo Method

3.1 The Filtering Algorithm

Algorithm 1 presents the XMC filtering method, which consists of three fundamental steps: simulation, fitting, and prediction. For conciseness, we assume that the state x_t is univariate; the multivariate case is discussed in Section 3.2. The simulation step ensures that N paths are available for the states and observations. The generated data are then split into two parts: a training sample, used as data in the regressions, and a validation sample, used for regularization of the regression method. When all regressions are completed, the states are predicted by evaluating the estimated regression functions with the actual observations.

We now consider the regularization step in more detail. Most regression methods involve tuning parameters, for which appropriate values must be chosen. This is accomplished by generating several candidate tuning parameters using a Bayesian optimization procedure (Bergstra, Yamins, & Cox, 2013) and selecting the best candidate based on the average loss for the validation sample. In practice it is usually sufficient to determine the tuning parameters at a single time point, t^*, for which suitable choices are discussed below.

We define the covariate set $\mathcal{C}_t$ to consist of the $W \in \{1, \ldots, T\}$ observations from the information set that are closest to time t, where the window size W is treated as a tuning

Algorithm 1 Extremum Monte Carlo method for signal extraction.

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

$$x_{1:T}^{(i)},\ y_{1:T}^{(i)}, \qquad i = 1, \dots, N.$$

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 regression at time $t = t^*$:

$$\widehat{f}_t^N \in \underset{f \in \mathbb{F}_N}{\arg\min} \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} L\left(x_t^{(i)} - f\left(\mathcal{C}_t^{(i)}\right)\right), \tag{4}$$

   with covariates $\mathcal{C}_t^{(i)} \subseteq \mathcal{F}_t^{(i)}$, loss function L, and function space $\mathbb{F}_N$. Select the tuning parameters that minimize the corresponding average loss for the validation sample.

   (c) *Regression*: Use the selected tuning parameters to perform the regression in (4) at all times $t = 1, \dots, T$ to obtain the function estimates $\{\widehat{f}_t^N\}_{t=1}^T$.

3. **Predict**: Evaluate the estimated regression functions with covariates based on the actual data, $\mathcal{C}_t$, to predict the states for $t = 1, \dots, T$:

$$\widehat{x}_t^N = \widehat{f}_t^N\left(\mathcal{C}_t\right).$$

parameter. For filtering, the information set is $\mathcal{F}_t = y_{1:t}$, and the covariate set is given by

$$\mathcal{C}_t = y_{s:t}, \qquad \text{with} \qquad s = s(t, W) = \max\{t - W + 1, 1\}, \tag{5}$$

so that $\mathcal{C}_t \subseteq \mathcal{F}_t$. For k-period forecasting, the covariate set is $\mathcal{C}_t = y_{s:t-k}$ with start index $s = \max\{t - k - W + 1, 1\}$, where $\mathcal{C}_t = \emptyset$ when $t - k < s$. For these prediction problems, $t^* = T$ is a natural choice in the regularization step, as it helps prevent underestimation of the window size because the validation loss is then computed where the maximum number of observations are available.

By specifying both the loss function and the regression method, Algorithm 1 defines a corresponding XMC filter. The loss function is required to have a bounded first moment, but otherwise, it can be chosen according to preference. The optimal regression method generally depends on the signal extraction problem. For the local level model, linear regression is suitable, while for other models, a nonlinear function estimator may be required. Selected regression methods that will be used for illustration include tree-based gradient boosting (GB; Friedman, 2001) and the random forest (RF; Breiman, 2001), where the latter can also be used for estimating conditional quantiles (Meinshausen, 2006). These methods were chosen for their general applicability and widespread use in practice. Furthermore, they remain applicable with discrete variables. However, Algorithm 1 can also be applied with other machine learning methods, such as neural networks. In the following, we discuss various features and extensions of XMC, which are illustrated numerically in Section 5.

3.2 Multivariate Filtering

A multivariate state $x_t = (x_{1,t}, \ldots, x_{N_x,t})'$ is handled by performing the fitting and prediction steps of Algorithm 1 separately for each element. This univariate treatment enables parallel computation across states, and it suggests that increasing the number of states N_x may have limited impact on the accuracy. To address large observation vectors y_t, tree-based regression methods (such as GB and RF) can be employed. These methods rely on selecting a subset of the most informative covariates, making them applicable in high-dimensional regression settings.

3.3 Steady State Filtering

A natural extension of Algorithm 1 is to reuse function estimates for prediction at other time points. We refer to this as the *steady state* XMC filter, by analogy to the Kalman filter (Durbin & Koopman, 2012, Ch. 4.3.4). The computational savings can be substantial for long time series, which are common in financial econometrics and the physical sciences.

The steady state approach allows us to stop performing regressions after some time t_{ss} and use the function estimate $\widehat{f}_{t_{\mathrm{ss}}}^{N}$ for prediction at subsequent time indices $t > t_{\mathrm{ss}}$. A minimal requirement for the steady state approach to be sensible is that the covariate sets correspond to rolling windows, such that after some time index $t_W \in \mathbb{N}$, we have

$$\mathcal{C}_{t+1} = \left\{ y_{j+1} \middle| y_j \in \mathcal{C}_t \right\} \qquad \forall\ t \geq t_W. \tag{6}$$

This condition means that the covariate set at time $t+1$ is obtained by applying the lead operator to every observation in $\mathcal{C}_t$. For filtering with covariate sets defined by (5), the above condition is satisfied with $t_W = W$. Any time $t \geq t_W$ is a feasible candidate for t_{ss}. Notably, if the steady state is invoked *a priori* at or before a specific time point τ (i.e., $t_{\text{ss}} \leq \tau$), it is only necessary to simulate paths of length τ. This allows the simulated data to be a fraction of the full sample size T in long time series, resulting in large computational gains.

Alternatively, a method for determining t_{ss} using the validation sample is as follows. For prudence, we rely on a conservative estimate of the predictive performance at subsequent time points, by evaluating a candidate regression function at time t based on prediction at time T, the most distant time point exceeding t. Let $c_{\text{ss}} \geq 0$ be a chosen tolerance level, and let the superscript $\langle i \rangle$ denote a case from the validation sample $(x_{1:T}^{\langle i \rangle}, y_{1:T}^{\langle i \rangle}),\ i = 1, \ldots, N_{\text{val}}$. Then, we verify for $t = t_W, \ldots, T-1$ whether

$$\sum_{i=1}^{N_{\text{val}}} L\left(x_T^{\langle i \rangle} - \widehat{f}_t^N \left(\mathcal{C}_T^{\langle i \rangle} \right) \right) \leq (1 + c_{\text{ss}}) \sum_{i=1}^{N_{\text{val}}} L\left(x_T^{\langle i \rangle} - \widehat{f}_T^N \left(\mathcal{C}_T^{\langle i \rangle} \right) \right) \tag{7}$$

is satisfied. When this condition is met, the iterative checking of (7) terminates, and we conclude that a steady state has been reached. We then set $t_{\text{ss}} := t$ and use the estimate $\widehat{f}_{t_{\text{ss}}}^N$ to circumvent the remaining regressions. This approach was applied with $c_{\text{ss}} = 0$ to compute the filtered states shown in Figure 1 (a). The condition in (7) was satisfied at time $t_{\text{ss}} = 19$, which allowed for circumventing 81% of the regressions.

3.4 Fixed-Interval Smoothing

In fixed-interval smoothing ("smoothing" hereafter), the information set is $y_{1:T}$, so that each state is estimated using all available data. The regularization step in Algorithm 1 is therefore performed at the middle time index $t^* = \lceil T/2 \rceil$, where the most nearby observations are available. Given a sufficient number of observations on both sides, the covariate set $\mathcal{C}_t$ corresponds to a centered window around time t. Otherwise, the window is asymmetric. An approach for constructing these covariate sets is described in Appendix A.

3.5 Missing Observations and Other Data Modifications

Several generalizations of the XMC method can be obtained by modifying the regression data. For example, the method can be extended to predict functions of the states, such as $g(x_t)$ or $g(x_{1:t})$, and to forecast future observations. This is done by adjusting the dependent variable accordingly in the regressions. Furthermore, missing observations can be handled by excluding the corresponding covariates from the regressions. This same approach can also be applied to handle data with unequal time spacing or vector measurements, where the elements are observed at varying frequencies.

3.6 Computational Complexity

To analyze the computational complexity of the XMC method, we focus on the regression step in Algorithm 1, which is generally the dominant runtime factor. The complexity is linear with respect to the number of states N_x and the time series length T, since separate regressions are performed for different states and times. When the steady state approach is used, the factor T is replaced by t_{ss}. The scaling in the number of paths N and covariates $C = WN_y$ depends on the selected regression method and its implementation. Table 1 presents current estimates of the computational complexity for several XMC filters.

Table 1: Computational complexity of the regression step for several XMC methods. The complexity for linear regression is based on ordinary least squares via the QR decomposition. For gradient boosting (GB), the estimate follows from the $O\big(CN\log(N)\big)$ complexity of a single regression tree with $C = WN_y$ covariates, while for the random forest (RF) it is based on the standard choice of $\sqrt{C}$ split variables (Hastie et al., 2009).

XMC Method	Linear	GB	RF
Complexity	$O\left(N_x T C^2 N\right)$	$O\big(N_x T C N \log(N)\big)$	$O\big(N_x T \sqrt{C} N \log(N)\big)$

4 Filter Stability and Convergence

This section discusses various theoretical properties of the XMC method. For ease of presentation, x_t is assumed to be univariate, which is without loss of generality due to the filter's separate treatment of the state elements. To avoid repetition, we focus on filtering, though most of the results also extend to forecasting and smoothing.

For the purpose of analysis, the XMC filter is viewed as a sequence of function estimators used for prediction,

$$\left\{\widehat{f}_t^N(\mathcal{C}_t)\right\}_{t=1}^{T}, \qquad \mathcal{C}_t \subseteq \mathcal{F}_t, \tag{8}$$

where the covariate set $\mathcal{C}_t$ is a subset of the information set $\mathcal{F}_t$ at each time t, and we use the following shorthand notation for the predictions:

$$\widehat{x}_t^N = \widehat{f}_t^N(\mathcal{C}_t).$$

The power set $\mathcal{P}(\mathcal{F}_t)$ represents the collection of all feasible covariate sets. Since the filter is fitted using a finite number of simulated paths N, the choice of covariate set (or window size) used in the regressions reflects a trade-off between the bias and variance of the filter.

Before proceeding, we present the following example to settle ideas.

Example (Linear XMC filter)**.** *Consider the XMC filter defined by using a linear regression function with parameters estimated by the least squares method,*

$$\widehat{f}_t^N(\mathcal{C}_t) = \sum_{y_j \in \mathcal{C}_t} \widehat{\beta}_{j,t} y_j, \tag{9}$$

for $t = 1, \ldots, T$, where we omit the intercept term for conciseness. Each feasible covariate set $\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)$ defines a different function estimator $\widehat{f}_t^N\big(\mathcal{C}_t\big)$. For instance, filtering at time $t = 2$ has information set $\mathcal{F}_t = \{y_1, y_2\}$, which gives three possible estimators:

$$\widehat{f}_2^N(\{y_1\}) = \widehat{\beta}_{1,2}^1 y_1, \qquad \widehat{f}_2^N(\{y_2\}) = \widehat{\beta}_{2,2}^2 y_2, \qquad \widehat{f}_2^N(\{y_1, y_2\}) = \widehat{\beta}_{1,2}^3 y_1 + \widehat{\beta}_{2,2}^3 y_2,$$

in addition to the trivial estimator $\widehat{f}_2^N(\emptyset) = 0$*. Any of these estimators could be used to predict the state* x_2*.*

4.1 Stability

As time progresses, the XMC method updates its predictions based on a moving window of observations. This contrasts with most filtering methods, where the predictions are defined recursively. Below we discuss several key properties of this approach.

The actual data, which serve as input for the predictions, are often characterized by extreme values or corrupted by errors. An important property implied by the use of moving windows is that the impact of any observation disappears once it drops from the window. Let $\eta \in \mathbb{R}$ denote a data error. The following result considers the case in which the measurement at some time $k \in \mathbb{N}$ is replaced by the corrupted measurement

$$y_k^\eta = y_k + \eta. \tag{10}$$

Corollary 1 (Finite window impact)**.** *Suppose the measurement at time* $k \in \mathbb{N}$ *is corrupted by a data error* $\eta \in \mathbb{R}$ *as in* (10)*. Let* $\widehat{x}_t^N(\eta)$ *be the resulting prediction of the XMC filter at time* t*, with covariate sets defined by* (5) *and window size* $W \in \mathbb{N}$*. Then*

$$\widehat{x}_t^N(\eta) = \widehat{x}_t^N(0) \quad \forall \quad t \geq k + W.$$

A related issue often encountered with recursive filters, such as the Kalman filter and its extensions, is that errors in the data or predictions can accumulate, causing the prediction error to grow over time. A filter which ensures that the error remains bounded over time, according to a suitable measure of distance, is said to possess the property of *stability* (e.g., Chopin & Papaspiliopoulos, 2020, Ch. 11.4). The following result provides sufficient conditions for the mean absolute error, $\mathbb{E}\,|x_t - \widehat{x}_t^N|$, to remain bounded over time, ensuring that the filter keeps track of the true state. It will be assumed that the states and measurements, (x_t, y_t), follow a strictly stationary process. This may differ from the SSM used in the simulation step of Algorithm 1, provided the process generating the simulated data is strictly stationary.

Theorem 1 (Filter stability)**.** *Let the process $\{(x_t, y_t)\}_{t\in\mathbb{N}}$ be strictly stationary, with $\mathbb{E}\,|x_1| < \infty$. For some fixed $N, W \in \mathbb{N}$, consider an XMC filter $\{\widehat{x}_t^N\}$ defined through Algorithm 1, with covariate sets given by (5), and assume that $\mathbb{E}\,|\widehat{x}_t^N| < \infty$ for $t \leq W$. Then, the mean absolute error remains bounded over time,*

$$\sup_{t\in\mathbb{N}} \mathbb{E}\,|x_t - \widehat{x}_t^N| < \infty, \tag{11}$$

where the expectation is with respect to the Monte Carlo draws and the actual data.

The above result remains applicable when the steady state approach from Section 3.3 is used and can be extended in several ways. The assumption that the covariate sets are defined via (5) can be replaced by the more general requirement that they satisfy the rolling window condition in (6), which also applies to k-period forecasting. Furthermore, the mean absolute error in (11) can be generalized to express the distance using the $\mathcal{L}^p$-norm $\|u\|_p = (\mathbb{E}\,|u|^p)^{1/p}$ for $p \geq 1$, with error $u = x_t - \widehat{x}_t^N$. The bounded moment requirements for the states and filter then pertain to the corresponding p-th moments.

4.2 Convergence

This section examines the convergence of the XMC filter to an optimal filter as the number of simulated paths N goes to infinity. The time series length $T \in \mathbb{N}$ is assumed to be finite, though it can be arbitrarily large. An optimal filter is characterized as follows.

Definition 1 (Optimal filter)**.** For a given SSM and loss function L, an *optimal filter* $\{f_t^*\}_{t=1}^T$ is a sequence of prediction functions satisfying

$$f_t^* \in \arg\min_f \mathbb{E}\left[L\{x_t - f(\mathcal{F}_t)\}\right], \qquad t = 1, \ldots, T. \tag{12}$$

For conciseness, the corresponding optimal filtered estimates will be denoted by

$$x_t^* = f_t^*(\mathcal{F}_t).$$

We will then say that the XMC filter converges in probability to an optimal filter when

$$\mathbb{E}_y \, |x_t^* - \widehat{x}_t^N| \xrightarrow{P} 0 \qquad \text{as} \qquad N \to \infty, \tag{13}$$

where $\mathbb{E}_y$ denotes the expectation with respect to the actual data, and the convergence in probability is with respect to the Monte Carlo draws. Similar results can be derived under under alternative modes of convergence, including pointwise convergence for the possible paths of the actual observations.

An important consideration in this context is that, while in the limit of $N \to \infty$ it is optimal to use the maximum number of covariates ($\mathcal{C}_t = \mathcal{F}_t$), in practice, this can lead to poor filtering performance due to high estimator variance, as only a finite amount of simulated data is available. In general, the optimal number of covariates in the regressions depends on N; see Appendix D.3 for an illustration. Therefore, it is important to choose it carefully. The XMC method addresses this by selecting the window size through minimization of the validation loss, which reflects the filter's predictive performance for the specific value of N used. To account for this essential aspect of Algorithm 1, the covariate set is allowed to depend on the simulated data, resulting in a regularized covariate set $\mathcal{C}_t^N$.

Definition 2 (Regularized covariate set)**.** Let $\mathcal{S}_N = \left\{ (x_{1:T}^{(i)}, y_{1:T}^{(i)})_{i=1}^N \right\}$ denote the set of all possible simulated samples of size N. A *regularized covariate set* $\mathcal{C}_t^N$ is a mapping that assigns each simulated sample $(x_{1:T}^{(i)}, y_{1:T}^{(i)})_{i=1}^N$ to a corresponding feasible covariate set:

$$\mathcal{C}_t^N \colon \mathcal{S}_N \to \mathcal{P}(\mathcal{F}_t).$$

In addition to the out-of-sample procedure in Section 3, the above definition accommodates other regularization methods, such as defining the window size as a deterministic function of N or incorporating a penalty term for the window size into the loss function.

Since the optimal filter generally relies on all available observations from the information set $\mathcal{F}_t$, a necessary requirement for establishing the convergence of the XMC filter is that the regularized covariate set converges to the information set when $N \to \infty$, as specified in the following assumption. Sufficient conditions to ensure its validity for several common regularization methods are discussed in Appendix C.

Assumption 1 (Convergence of regularized covariate sets)**.** *For $t = 1, \ldots, T$ the regularized covariate set converges in probability to the corresponding information set,*

$$\lim_{N\to\infty} P\Big(\mathcal{C}_t^N = \mathcal{F}_t\Big) = 1,$$

where the convergence is with respect to the Monte Carlo draws.

The following result provides sufficient conditions for convergence of the linear XMC filter to the Kalman filter in the important special case of linear Gaussian SSMs.

Theorem 2 (Convergence to Kalman filter)**.** *Let Assumption 1 hold, and suppose the following holds:*

A2.1 The SSM in (1) *is linear and Gaussian, where the initial state $x_1 \sim \mathrm{N}(0, \Sigma_1)$ has bounded variance and is independent of the noise terms $\{\varepsilon_t^x\}$ and $\{\varepsilon_t^y\}$, which are mutually and serially independent. The observations in $y_{1:T}$ are linearly independent.*

A2.2 L is the squared error loss.

A2.3 The XMC filter is linear as in (9)*.*

Then, the XMC filter converges in probability to the Kalman filter $\{x_t^*\}$ at rate $\sqrt{N}$,

$$\sup_t \sqrt{N}\, \mathbb{E}_y \left| x_t^* - \widehat{x}_t^N \right| = O_P(1).$$

In Theorem 2, the assumption that the observations are linearly independent holds for most SSMs of practical interest, with a sufficient condition being that the measurement noise is non-degenerate. Assumption A2.2 can be relaxed, for instance, to accommodate the absolute error loss via Theorem 2 of Pollard (1991). Moreover, the XMC filter can rely on nonlinear regression methods.

The latter is particularly relevant for nonlinear and non-Gaussian SSMs, where the functional form of the optimal filter is usually unknown. In such settings, nonparametric estimators may offer a solution, as this class contains a large number of alternatives that guarantee convergence under mild assumptions. For example, Zhang and Yu (2005) present convergence rates for general boosting procedures with early stopping, Peng, Coleman, and Mentch (2022) provide results for the RF method, and van de Geer (2000) and Chen (2007) give overviews of convergence rates for (semi-)nonparametric methods, with the latter discussing several common variants of neural networks.

A separate analysis for the above alternatives is beyond the scope of this paper. Instead, we provide a general auxiliary result to simplify the process of establishing convergence. The result shows that if the chosen regression functions result in convergence to the optimal filter when all possible covariates are used ($\mathcal{C}_t = \mathcal{F}_t$), Assumption 1 ensures that the consistency and convergence rate of the estimators are unaffected by the use of a regularized covariate set. Consequently, the regularized covariate set can be disregarded in convergence analyses.

Lemma 1 (General filter convergence)**.** *Suppose that for the given SSM and loss function, there exists an optimal filter $\{x_t^*\}$ as specified in Definition 1. Let Assumption 1 hold, and let $r_N \, \mathbb{E}_y \left| x_t^* - \widehat{f}_t^N(\mathcal{F}_t) \right| = O_P(1)$ for $t = 1, \ldots, T$ with rate $r_N > 0$ that diverges as $N \to \infty$. Then, the XMC filter converges in probability to the optimal filter at rate r_N,*

$$\sup_t r_N \, \mathbb{E}_y \left| x_t^* - \widehat{x}_t^N \right| = O_P(1).$$

Lemma 1 shows that the XMC filter retains the asymptotic properties of the regression method it uses. In this context, we note that the lack of tractability in nonlinear and non-Gaussian models makes it challenging to design parametric estimators that converge to the optimal filter. The practical implication is that general regression methods (such as GB and RF) play an important role in broadening the applicability of the XMC method.

5 Applications

This section presents various applications to illustrate the key properties of the XMC method. We set the validation sample fraction to $c_{\text{val}} = 0.1$. Additional applications can be found in Appendix D.

5.1 Non-Markovian Models

Applications involving non-Markovian models are common, typically arising from higher-order autoregressive state processes or serially dependent noise. For example, by modifying the local level model in (3) to let the state noise follow a random walk, we obtain the integrated random walk plus noise model (e.g., Durbin & Koopman, 2012, Ch. 3),

$$\begin{aligned} y_t &= x_t + \varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{N}(0, \sigma_y^2), \\ x_{t+1} &= x_t + \varepsilon_t^x, & & \\ \varepsilon_{t+1}^x &= \varepsilon_t^x + u_t, & u_t &\sim \mathrm{N}(0, \sigma_u^2), \end{aligned} \tag{14}$$

for $t = 1, \ldots, T$, where the noise sequences $\{u_t\}$ and $\{\varepsilon_t^y\}$ are mutually and serially independent, and approximate diffuse initializations are used for x_1 and ε_1^x. We can treat the non-Markovian structure of the model in (14) using an augmented state vector, in this case $z_t = (x_t, \varepsilon_t^x)'$, to restore the Markov property. The resulting SSM is given by

$$\begin{aligned} y_t &= \begin{pmatrix} 1 & 0 \end{pmatrix} z_t + \varepsilon_t^y, \\ z_{t+1} &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} z_t + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u_t. \end{aligned} \tag{15}$$

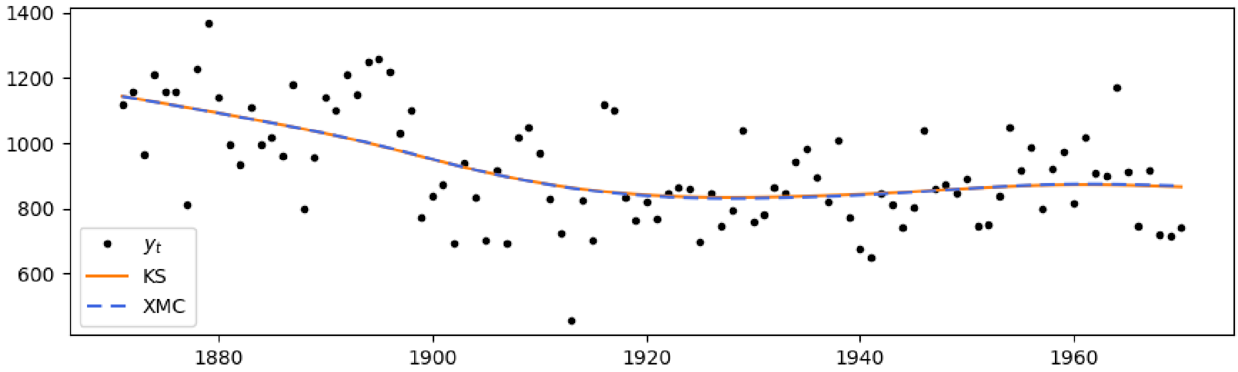


Figure 2: Smoothed signals for the Nile data based on the integrated random walk plus noise model in (14): signals extracted via $\mathbb{E}[x_t|y_{1:T}]$ by the linear XMC smoother with $N = 10^5$, as well as the Kalman smoother (KS) applied to the augmented state form of the model in (15).

An important property of the XMC method is its direct applicability to non-Markovian models. To illustrate this feature, we consider the model in (14) for the Nile data. We set the static parameters to the maximum likelihood estimates $\sigma_u = 1.276$ and $\sigma_y = 137.741$, with approximate diffuse initializations. Figure 2 shows the smoothing means $\mathbb{E}[x_t|y_{1:T}]$ computed via the Kalman smoother (Anderson & Moore, 1979, Ch. 7) based on the augmented SSM given by (15). The plot also shows the corresponding estimates by the linear XMC method with $N = 10^5$, based on the original model in (14). The estimates match the exact smoothing means.

The above example illustrates that the XMC method can be useful in the analysis of non-Markovian models. The ability to treat such models directly could be especially helpful in cases where large augmented state vectors are needed. For instance, software implementations of the popular ARIMA(p, d, q) model typically adopt a companion form that requires a state vector of size $d + \max\{p, q + 1\}$; see Durbin and Koopman (2012, Ch. 3.4). Depending on the chosen specification, the augmented state vector may be considerably larger than the $N_x = 1$ case handled by the XMC method. This convenient feature of XMC can lead to substantial savings in both computation and storage.

5.2 Nonlinear Signal Extraction

This section considers signal extraction for the univariate nonlinear model given by

$$\begin{aligned} y_t &= \frac{x_t^2}{20} + \varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{N}(0, \sigma_y^2), \\ x_{t+1} &= \frac{1}{2} x_t + \frac{25 x_t}{1 + x_t^2} + 8 \cos\big(1.2(t+1)\big) + \varepsilon_t^x, & \varepsilon_t^x &\sim \mathrm{N}(0, \sigma_x^2), \end{aligned} \tag{16}$$

where $x_1 \sim \mathrm{N}(0, 1)$. This model is often used for testing the performance of nonlinear filtering and smoothing methods (Gordon et al. 1993; Kitagawa 1996; Godsill, Doucet, and West 2004). We set the static parameters to $\sigma_x^2 = 0.1$ and $\sigma_y^2 = 1$ as in Kitagawa (1996).

5.2.1 The Importance of General Regression Methods

Figures 3 (a) and (b) present a simulated path of the observations $y_{1:T}$ and states $x_{1:T}$, respectively. Estimates of the corresponding smoothing means from the GB-XMC smoother with $N = 10^5$ are shown in Figure 3 (b). For comparison, Figure 3 (b) also includes estimates of the smoothing means based on particle smoothing, where the forward filter backward smoothing (FFBS; Doucet, Godsill, & Andrieu, 2000) method was applied with 10^4 particles from the auxiliary particle filter (APF; Pitt & Shephard, 1999) to ensure high accuracy of the estimates. The estimates from GB-XMC and FFBS are visually identical.

Figure 3 (c) presents the differences with the FFBS estimates for several XMC smoothers. By comparing the scales of Figures 3 (b) and (c), we find that the nonlinear regression methods (GB and RF) lead to an adequate performance of the XMC smoother. In contrast, the linear smoother (Lin-XMC) does not provide satisfactory performance, which is unchanged when the number of draws increases. This implies that the smoothing means $\mathbb{E}\left[x_t | y_{1:T}\right]$ are inherently nonlinear in the observations, and it demonstrates the importance of general regression methods for nonlinear signal extraction with the XMC method.

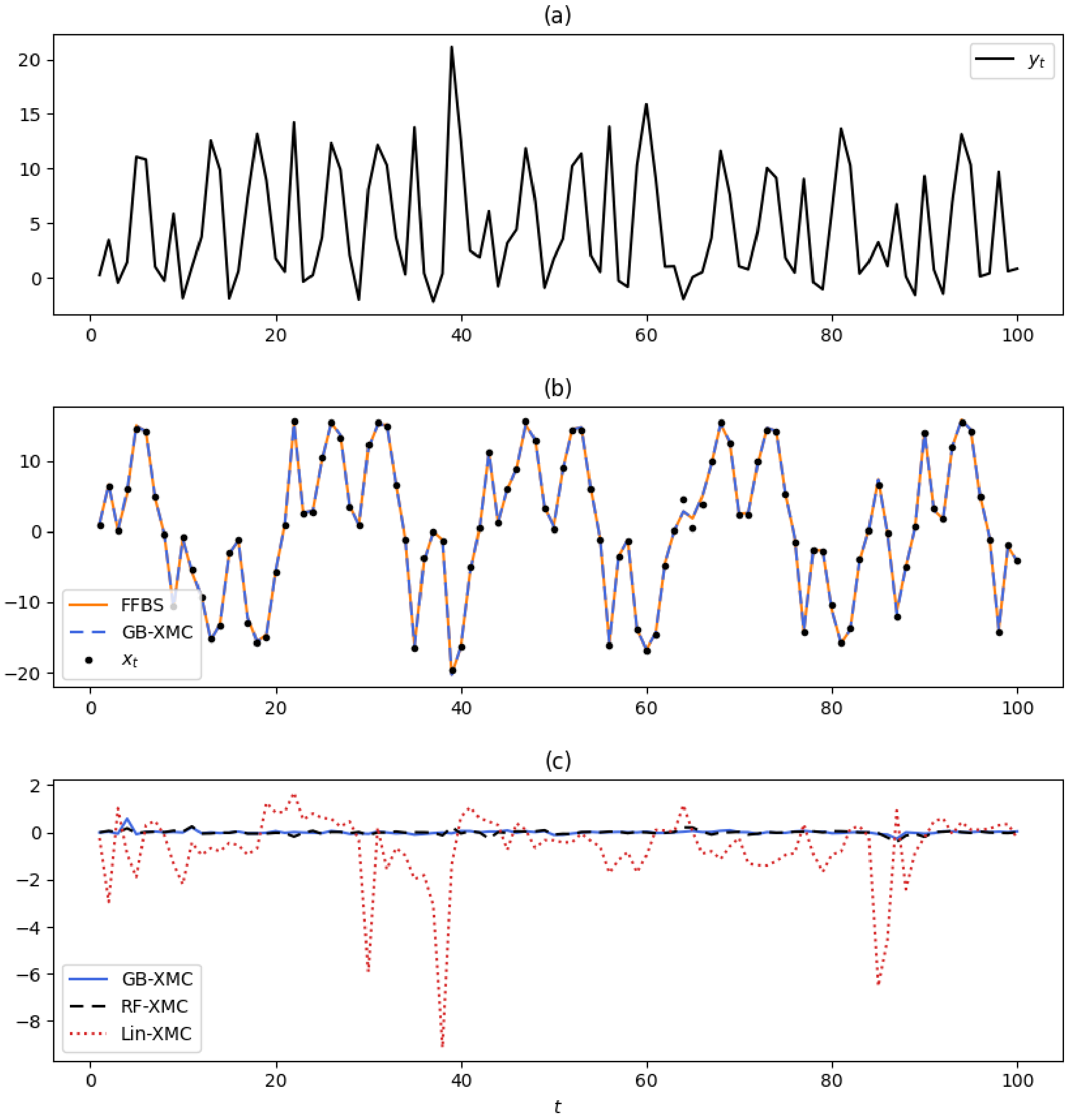


Figure 3: Analysis of a simulated path from the nonlinear model in (16): (a) observations; (b) true and smoothed states based on estimated smoothing means computed via the forward filter backward smoothing (FFBS) method with 10^4 particles from the auxiliary particle filter, and the gradient boosting (GB) XMC smoother with $N = 10^5$; (c) differences with FFBS for the GB, random forest (RF), and linear (Lin) XMC smoothers.

5.2.2 Simulation Study

The XMC method can be applied with a loss function of choice, enabling the estimation of time-varying conditional means, medians, and other quantities of interest. To illustrate this feature, we conducted a simulation study based on the nonlinear model in (16), using the absolute error loss to estimate the smoothing medians and the squared error loss for the

Table 2: Results from smoothing simulation study based on 10^3 test paths of length $T = 100$ from the nonlinear model in (16): overall mean absolute error (MAE) based on the random forest XMC smoother for quantile regression and root mean squared error (RMSE) based on the gradient boosting XMC smoother for various values of N. The final row displays the excess loss as a percentage relative to the FFBS method's performance using $M = 10^4$ particles, which achieved an MAE of 0.342 (for the medians) and an RMSE of 0.562 (for the means).

$\log_{10}(N)$	3		4		5	
Error metric	MAE	RMSE	MAE	RMSE	MAE	RMSE
Performance	0.405	0.774	0.360	0.594	0.349	0.571
Relative (Excess %)	18.4	37.7	5.3	5.7	2.0	1.6

smoothing means. Specifically, the nonlinear model was used to simulate 10^3 test paths of length $T = 100$, after which the states were predicted using the corresponding observations. The smoothing medians were estimated using the RF-XMC method for quantile regression, while the smoothing means were estimated using the GB-XMC method.

The first row ("Performance") in Table 2 shows the overall mean absolute error (MAE)—defined as the absolute error over over the times and paths—based on the estimated medians for various values of N. In addition, the overall root mean squared error (RMSE) is provided for the estimated means, which is computed as the square root of the overall mean squared error. The initial improvements from $N = 10^3$ to $N = 10^4$ are substantial, while the increase to $N = 10^5$ results in smaller gains. This illustrates that general regression methods like GB and RF tend to require more data to perform adequately. Notably, as the XMC method uses simulated data, N is limited only by computational constraints.

As a proxy for the optimal smoother, we also estimated the smoothing means and medians using the FFBS method with $M = 10^4$ particles from the APF. This resulted in an MAE of 0.342 (for the medians) and an RMSE of 0.562 (for the means). The value of M is substantial, given the computational complexity of $O(TM^2)$ for the FFBS method, which is typical for particle smoothing methods (Chopin & Papaspiliopoulos, 2020, Ch. 12). The final row ("Relative") presents the excess loss of the XMC smoothing methods as a percentage relative to the FFBS method. The use of $N = 10^5$ yields adequate relative performance, with excess losses of about 2% compared to the optimal smoother proxy.

A direct comparison between M and N is complicated. For several reasons, the simulated paths are less costly than particles, with the precise costs depending on the choice of importance sampler and XMC smoother. On the other hand, fewer particles are needed to achieve a given level of accuracy, as these draws are conditional on the actual data, whereas the simulated paths are not. Regarding the computational complexity of the XMC smoothing method, Table 1 shows that the linear version has complexity $O(TN)$, while the GB and RF-XMC smoothers have complexity $O\big(TN\log(N)\big)$. It is worth noting that the latter is $O(TN^{1+\delta})$ for any $\delta > 0$, so depending on the type of regression, the method scales either linearly or almost linearly with N. This allows large values of N to be used in practice, enabling accurate smoothing estimates with the XMC method.

5.3 Real-Time Filtering

For real-time filtering problems, an important property of the XMC method is that most of the computations take place in the simulation and fitting steps, which can be performed offline. To illustrate this, we performed a filtering simulation study based on the nonlinear model in (16). The model was used to simulate 10^3 test paths of length $T = 100$, and for each path the simulated states were predicted using the corresponding observations. The GB-XMC filter was used to estimate the filtering means $\mathbb{E}[x_t|y_{1:t}]$. Table 3 presents the results of the simulation study. As with smoothing, the RMSE drops substantially when N increases from 10^3 to 10^4, followed by a more moderate improvement for $N = 10^5$.

As a proxy for the optimal filter, we also applied the APF with 10^5 particles, which achieved an RMSE of 1.645. This comparison indicates that the XMC filter performs adequately starting at $N = 10^4$, where the excess RMSE relative to the APF is 3.2%. For $N = 10^5$, the performance further improves, yielding an excess RMSE within one percent.

Table 3 also shows the total computation times in seconds, divided into offline and online calculations. The offline times correspond to the regression step in Algorithm 1, while the online times correspond to the prediction step. By increasing the number of simulated paths N, the offline computation time increases. In contrast, the online computation

Table 3: Results from filtering simulation study based on 10^3 test paths of length $T = 100$ from the nonlinear model in (16): overall root mean squared error (RMSE) and computation times in seconds based on the gradient boosting XMC filter for various values of N. The second row ("Excess RMSE") displays the excess loss as a percentage relative to the performance of the auxiliary particle filter using 10^5 particles, which achieved an RMSE of 1.645. Offline computation times correspond to the regression step and online times to the prediction step in Algorithm 1, based on a computer with AMD Ryzen 5-5500U processor.

$\log_{10}(N)$		3	4	5
RMSE		1.850	1.697	1.659
Excess RMSE (%)		12.5	3.2	0.9
Time (s)	Offline	11.9	139.9	1194.0
	Online	0.6	0.6	0.7

times, which account for a small fraction of the total times, remain largely unaffected, as the online phase consists of evaluating functions that have been pre-estimated. Given that the computational bottleneck in Algorithm 1 can be handled offline, the XMC filtering method could offer an accurate solution for real-time problems.

5.4 Multivariate Filtering: Linear Gaussian Model

For high-dimensional models, signal extraction faces challenges due to the need to evaluate high-dimensional densities. In the case of particle filters, this leads to the well-known issue of weight degeneracy (Snyder, Bengtsson, Bickel, and Anderson 2008; Snyder, Bengtsson, and Morzfeld 2015; Chopin and Papaspiliopoulos 2020, Ch. 19.1). To investigate the performance of the XMC method in high-dimensional settings, we consider the linear Gaussian model given by

$$\begin{aligned} y_t &= x_t + \varepsilon_t^y, \qquad && \varepsilon_t^y \sim \mathrm{N}(0, I_d), \\ x_{t+1} &= 0.9 x_t + \varepsilon_t^x, \qquad && \varepsilon_t^x \sim \mathrm{N}(0, R), \end{aligned} \tag{17}$$

with $x_1 \sim \mathrm{N}\big(0, (1-0.9^2)^{-1}R\big)$, where $x_t, y_t, \varepsilon_t^x, \varepsilon_t^y$ are $d \times 1$ vectors for some $d \in \mathbb{N}$ and R is a $d \times d$ variance matrix. This model has the well-known structure of a seemingly unrelated time series equations model and serves as a convenient test case for two reasons. First, it allows for investigating the performance of the XMC filtering method across different model dimensions d. Second, the linear Gaussian structure enables a comparison with the

Table 4: Results from simulation study based on the multivariate linear Gaussian model in (17) for different values of the model dimension d: overall root mean squared error (RMSE) from 10^4 test paths of length $T = 50$ for the gradient boosting XMC filter with various values of N. The second row ("Excess RMSE") displays the excess loss as a percentage relative to the performance of the Kalman filter, which achieved an RMSE of 0.740 in each case.

d	25			50			100		
$\log_{10}(N)$	3	4	5	3	4	5	3	4	5
RMSE	0.815	0.761	0.747	0.809	0.767	0.748	0.816	0.765	0.748
Excess RMSE (%)	10.1	2.8	0.9	9.3	3.6	1.1	10.3	3.4	1.1

exact filtering means, which can be computed using the Kalman filter.

The relationship between elements of the observations in (17) is governed by the state noise variance matrix R. This matrix is chosen so that all states are equally difficult to predict, facilitating comparison and aggregation of their prediction errors. To achieve this, we use a circular distance function, $D(i, j)$, which ensures that state index 1 is equidistant from both index 2 and the final index $N_x = d$. Specifically, we define the elements of R as

$$R_{i,j} = 0.5^{D(i,j)}, \qquad D(i, j) = \min\{|i - j|, d - |i - j|\}, \tag{18}$$

for $i, j = 1, \ldots, d$. Thus, R is a correlation matrix, and the signal-to-noise ratio equals one.

A simulation study is performed with the model in (17) for dimensions $d \in \{25, 50, 100\}$. For 10^4 test paths of length $T = 50$, the GB-XMC filter is applied to estimate the filtering means $\mathbb{E}[x_t|y_{1:t}]$ with various values of N. Table 4 presents the overall RMSE values. With the model dimension d held fixed, the RMSE decreases substantially from $N = 10^3$ to $N = 10^4$, and more gradually to $N = 10^5$, as in the previous simulation studies. When N is held fixed, increasing the model dimension has little impact on the accuracy.

The final row ("Excess RMSE") in Table 4 presents the excess loss of the XMC filter as a percentage relative to the performance of the Kalman filter, which achieved an RMSE of 0.740 in each case. The XMC filter shows adequate relative performance for $N = 10^4$, with an excess RMSE of at most 3.6% (for $d = 50$). The accuracy further improves for $N = 10^5$, resulting in an excess RMSE of approximately one percent relative to the optimal filter.

5.5 Multivariate Filtering: Stochastic Volatility Model

We consider the multivariate stochastic volatility (MSV) model from Harvey, Ruiz, and Shephard (1994) given by

$$\begin{aligned} y_t &= \exp(0.5x_t) \odot \varepsilon_t^y, & \varepsilon_t^y &\sim \mathrm{N}(0, R), \\ x_{t+1} &= \mu + \Phi(x_t - \mu) + \varepsilon_t^x, & \varepsilon_t^x &\sim \mathrm{N}(0, \Sigma), \end{aligned} \tag{19}$$

where $x_t, y_t, \varepsilon_t^x, \varepsilon_t^y$ are all $d \times 1$ vectors for some $d \in \mathbb{N}$, the operator $\odot$ denotes elementwise multiplication, R is a $d \times d$ correlation matrix, μ is a $d \times 1$ vector, and Σ is a $d \times d$ variance matrix. The vector y_t represents the log returns of d financial assets, and x_t contains their corresponding log variances. The above model is standard in multivariate volatility modeling and is often called the basic MSV model (Asai, McAleer, & Yu, 2006; Chib, Omori, & Asai, 2009).

5.5.1 Simulation Study

We conducted a simulation study using the MSV in (19) for dimensions $d \in \{25, 50, 100\}$, with parameters $\Phi = 0.9I_d$, $\Sigma = I_d$, $\mu = 0$, and the correlation matrix R as defined in (18). The initial state follows $x_1 \sim \mathrm{N}\big(0, (1 - 0.9^2)^{-1} I_d\big)$. For 10^4 test paths of length $T = 50$, the GB-XMC method is used to estimate the filtering means $\mathbb{E}[x_t | y_{1:t}]$ and forecasting means $\mathbb{E}[x_t | y_{1:t-1}]$ with various values of N. Table 5 presents the results of the simulation study. For both filtering and forecasting at any fixed model dimension d, the RMSE shows a large improvement from $N = 10^3$ to $N = 10^4$, and a smaller improvement for $N = 10^5$. Keeping N fixed, the RMSE remains almost unchanged when the model dimension d increases.

In practice, the quasi-maximum likelihood (QML) method of Harvey et al. (1994) is often used for estimating the states. This approach involves linearizing the model via the transformation $\widetilde{y}_{i,t} = \log(y_{i,t}^2)$ for $i = 1, \ldots, d$, and using the Kalman filter to estimate the states, which results in the minimum variance linear unbiased estimator of x_t based on the transformed measurements. The XMC method also offers a straightforward approach for

Table 5: Results from simulation study based on the MSV model in (19) for different values of the model dimension d: overall root mean squared error (RMSE) for filtering and 1-period forecasting applied to 10^4 test paths of the states with length $T = 50$. The results are presented for the gradient boosting XMC filter using various values of N, and the quasi-maximum likelihood (QML) method of Harvey et al. (1994).

d		25			50			100		
$\log_{10}(N)$		3	4	5	3	4	5	3	4	5
RMSE (Filtering)	XMC	1.241	1.134	1.095	1.258	1.136	1.097	1.298	1.140	1.098
	QML		1.253			1.253			1.253	
RMSE (Forecasting)	XMC	1.564	1.467	1.433	1.571	1.470	1.433	1.590	1.470	1.435
	QML		1.528			1.527			1.527	

estimating the unobserved states, but it is based on the actual MSV model in (19), hence it is of interest to compare the performance of these methods. Table 5 shows that, starting at $N = 10^4$, the XMC method outperforms the QML method in both filtering and forecasting. Further improvement is observed for $N = 10^5$, where the performance differences are substantial. These findings are consistent across the different model dimensions.

5.5.2 Parameter estimation

In empirical applications, the XMC method can be paired with various approaches to parameter estimation. For nonlinear and non-Gaussian models, it is often possible to derive expressions for moments of (functions of) the data, which has led to the widespread use of the generalized method of moments (Hansen, 1982) in econometrics. In the next section, we estimate the parameters of the MSV model in (19) via the moment-based approach of Ahsan and Dufour (2024). This approach applies the QML transformation to the data and derives corresponding moments, which can be inverted analytically to obtain a closed-form estimator. The estimator is consistent and asymptotically normal under suitable regularity conditions and does not require the common assumptions of diagonal matrices for Φ, Σ, and R, allowing us to estimate the full model.

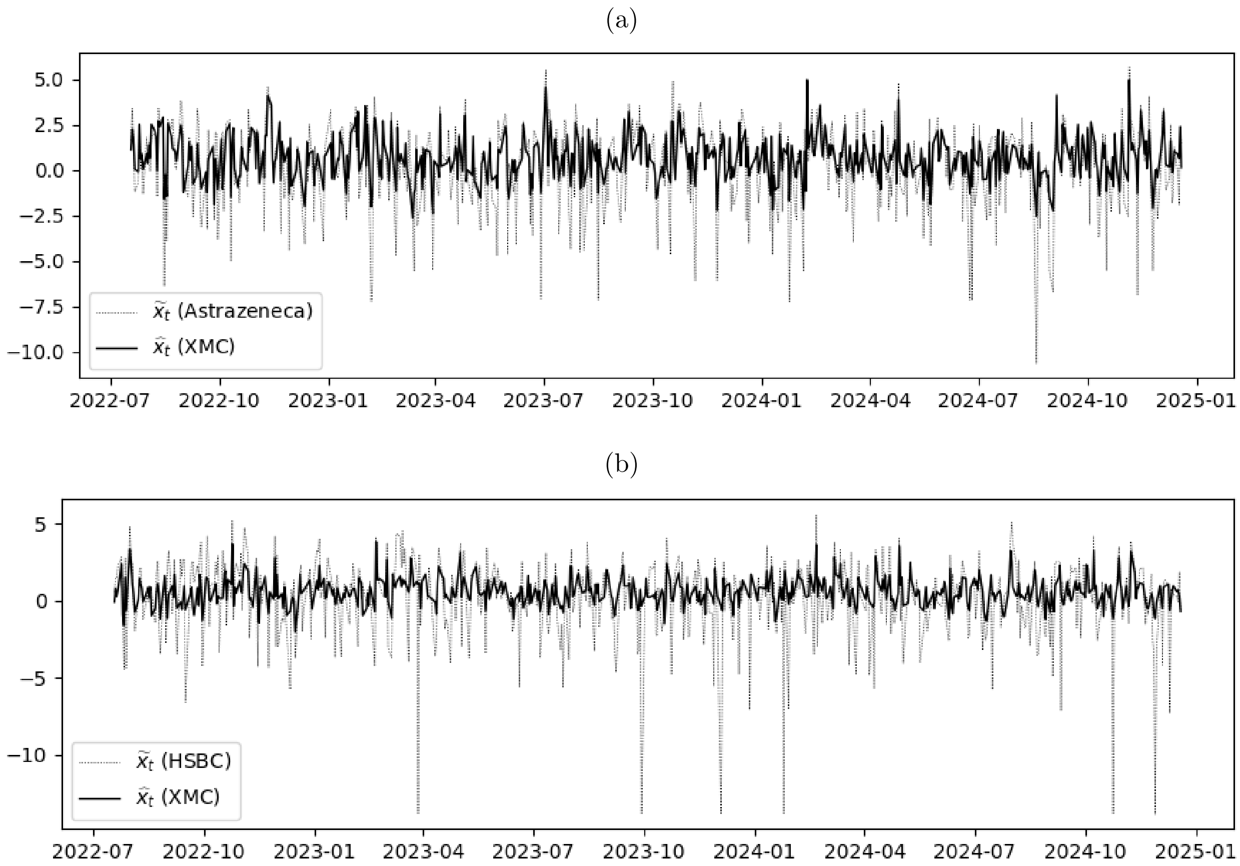
When analytical expressions for the moments are unavailable, simulation-based estimators can be employed (e.g., Gouriéroux & Monfort, 1996). A widely used example is the method of simulated moments (McFadden, 1989), in which the moments are estimated using simulated samples from the model. The indirect inference estimator (Gourieroux, Monfort, & Renault, 1993) generalizes this approach by minimizing the distance between simulated and actual data, using an auxiliary model that is similar to, but easier to estimate than, the model of interest. The XMC method integrates naturally with these estimators in complex settings, with all three approaches relying on simulated samples from the model.

5.5.3 Empirical Application: The FTSE 100 Index

This section applies the XMC filter to a high-dimensional stochastic volatility model, using a daily time series of the constituents from the Financial Times Stock Exchange 100 (FTSE 100) Index. This index represents the 100 largest companies by market capitalization listed on the London Stock Exchange and is widely regarded as a leading economic indicator. Our analysis employs the MSV model in (19), where y_t represents a 100×1 vector containing the daily log returns of the index constituents, and with x_t the corresponding log variances. We estimate the parameters using the longest available sample for each stock, with data for some constituents going back as far as December 2004. The combined sample ends on 19 December 2024. The data for this analysis were obtained using *LSEG Workspace* (London Stock Exchange Group, 2025).

We focus on signal extraction for the recent period from July 19$^{\text{th}}$, 2022 until December 19$^{\text{th}}$ 2024. The analysis is based on the MSV model of dimension 100, but the univariate treatment allows for estimating only those states that are of interest to the analyst. As illustration, Figure 4 shows the estimated filtering means of x_t for AstraZeneca (ticker: AZN) and HSBC (ticker: HSBA), two of the largest FTSE 100 constituents by market capitalization. The estimates are obtained using the steady state GB-XMC filter with $N = 10^5$. To save computations, the steady state time t_{ss} was set *a priori* to the earliest feasible index implied by the condition in (6). This resulted in $t_{\text{ss}} = 5$ for AstraZeneca and $t_{\text{ss}} = 6$ for HSBC, which enabled the circumvention of more than 99% of the regressions in Algorithm 1.

To validate the filtered states relative to the data, we first apply the QML transformation to the log returns, $\widetilde{y}_t = \log(y_t^2) = x_t + \log(|\varepsilon_t^y|^2)$. Next, we adjust the mean of these


(a)

(b)

Figure 4: Analysis of the log returns of FTSE 100 index constituents based on the MSV model in (19): filtered states $\widehat{x}_t \approx \mathbb{E}[x_t|y_{1:t}]$ based on the gradient boosting XMC filter with $N = 10^5$ and state proxies $\widetilde{x}_t$ as defined in (20) for AstraZeneca (a) and HSBC (b).

transformed data to obtain the state proxies: $\widetilde{x}_t = \widetilde{y}_t - \mathbb{E}\left[\log(|\varepsilon_t^y|^2)\right]$. It follows that

$$\widetilde{x}_t = x_t + \xi_t, \tag{20}$$

where $\xi_t = \log(|\varepsilon_t^y|^2) - \mathbb{E}\left[\log(|\varepsilon_t^y|^2)\right]$ is white noise. Figure 4 presents the state proxies alongside the filtered states, showing that they are generally updated in the same direction. However, the filter is much less sensitive to extreme observations. In this context, we note that the use of GB (and other tree-based regression methods) ensures a bounded impact of outliers due to flat extrapolation. Furthermore, Corollary 1 shows that the impact of any observation vanishes once it falls outside the moving window used by the XMC filter.

6 Discussion

This section discusses related work and extensions of the XMC method. First, we note that the XMC treatment of non-Markovian models readily extends to smoothing in nonlinear and non-Gaussian settings, where particle smoothing methods are often employed (Chopin & Papaspiliopoulos, 2020, Ch. 12). Most variants (e.g., FFBS) combine particles from a forward filter with a backward pass, using the state transition density to weight particle pairs at times t and $t+1$. However, in non-Markovian models—where an augmented state vector z_t is required—the transition density $p\left(z_{t+1}^{(j)}|z_t^{(i)}\right)$ leads to degenerate weights when combining particles from different paths $(i \neq j)$, since part of z_{t+1} is fixed given its lag z_t. A detailed discussion of this matter is provided by Fearnhead, Wyncoll, and Tawn (2010). In this setting, the XMC method offers a natural solution, as it can be applied directly to non-Markovian models.

In our discussion of smoothing, we have focused on the fixed-interval variant, as it is the most commonly used form in economics. However, the XMC method could be extended to other forms of smoothing, such as fixed-lag and fixed-point smoothing (Anderson & Moore, 1979, Ch. 7). This extension will be explored in a future study.

The XMC method could also be extended to accommodate the estimation of conditional distributions by letting the loss function have the more general form $L(x_t, \mathcal{C}_t, f)$, where $f = f(x_t|\mathcal{C}_t)$ is a conditional density or probability mass function. By using the observations y_t as dependent variable and covariates $\mathcal{C}_t \subseteq y_{1:t-1}$, this approach could be used to estimate the likelihood via the prediction decomposition $p(y_{1:T};\theta) = p(y_1;\theta)\prod_{t=2}^{T} p(y_t|y_{1:t-1};\theta)$, where θ is a parameter vector. This idea is related to the reprojection method of Gallant and Tauchen (1998), in which a long simulated path of the observations is used to perform maximum likelihood via estimates of the observation transition density. It is also connected to the Bayesian amortized inference approach (Stuhlmüller, Taylor, & Goodman, 2013), in which draws from the prior are used to train a neural network approximation to the posterior density. The XMC method is also linked to amortized inference through the steady state approach described in Section 3.3, where the costs of estimation are amortized by reusing the regression functions over time.

The above extension can be useful for model comparison, maximum likelihood estimation, and hypothesis testing. It could also be used to extract point estimates from the estimated distributions. However, a key advantage of the current approach is that it is generally more efficient—statistically and computationally—to directly estimate the prediction functions of interest. A related point is made by Raynal et al. (2019) in the context of approximate Bayesian computation (ABC) for inference on static parameters θ. They focus on estimating posterior moments to automate the use of summary statistics, by performing regressions of the static parameters drawn from a prior distribution $p(\theta)$ on a large number of summary statistics using the RF method. The RF method is considered pivotal to their approach because its robustness to irrelevant predictors enables bypassing the usual preliminary selection of the summary statistics (Raynal et al., 2019, p. 1722).

The XMC method contrasts with their approach and other simulation-based methods for estimating static parameters, such as indirect inference and the simulated method of moments, as it is specifically designed for signal extraction in the context of general SSMs. This introduces several distinctive elements that are outside the scope of the static setting, such as the steady state approach, the construction of suitable covariate sets for different information sets, the finite window impact of observations, and filter stability. Another key distinction from the above and other ABC approaches is that the XMC method does not rely on summary statistics. Instead, data reduction is achieved through the covariate sets, which consist of the observations nearest in time to the signal of interest. As shown in Lemma 1, this allows for establishing convergence by means of a sufficiently general regression method. Furthermore, the XMC filtering algorithm is flexible with respect to the choice of regression method. This enables the use of linear regression, (gradient) boosting, neural networks, and other machine learning methods. Likewise, XMC can be paired with a loss function of choice, offering flexibility for a wide range of prediction tasks in signal extraction.

7 Conclusion

This paper introduces a novel simulation-based method for signal extraction in a general class of SSMs. The XMC method can be used when the model allows for simulation and its implementation is straightforward. In complex settings, it combines naturally with the method of simulated moments and indirect inference for parameter estimation. For applications where a Bayesian or maximum likelihood estimator is adopted, the XMC method offers a fast solution for real-time filtering. In addition, the steady state approach can provide an efficient means of signal extraction in applications with long time series.

Conditions for the stability and convergence of the filtering method are discussed, and the key properties of the method are illustrated through various applications, including nonlinear and high-dimensional models. The results indicate that state estimation in nonlinear and non-Gaussian models can be performed effectively using regression—a technique well known to applied economists—with simulated data. In this way, the XMC method presents an accessible approach to nonlinear signal extraction.

Acknowledgments

This paper was presented at the 2023 NESG Seminar in Rotterdam and the 2024 NBER-NSF Time Series Conference in Philadelphia. The authors are grateful for comments provided by Neil Shephard, Simon Godsill, Nalan Bastürk, Frank van der Meulen, Lennart Hoogerheide, Rong Chen, James Hamilton, and other conference participants.

Funding Details

This work was supported by the Dutch Research Council (Grant VI.Vidi.195.099; Blasques).

Disclosure Statement

The authors report there are no competing interests to declare.

References

Ahsan, N., & Dufour, J.-M. (2024). Practical estimation methods for high-dimensional multivariate stochastic volatility models. *Available at SSRN 5081221*.

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

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

Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *72*(3), 269–342.

Asai, M., McAleer, M., & Yu, J. (2006). Multivariate stochastic volatility: a review. *Econometric Reviews*, *25*(2-3), 145–175.

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).

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

Chen, X. (2007). Large sample sieve estimation of semi-nonparametric models. *Handbook of Econometrics*, *6*, 5549–5632.

Chib, S., Omori, Y., & Asai, M. (2009). Multivariate stochastic volatility. In *Handbook of financial time series* (pp. 365–400). Springer.

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

Cobb, G. W. (1978). The problem of the Nile: Conditional solution to a changepoint problem. *Biometrika*, *65*(2), 243–251.

Creal, D. (2012). A survey of sequential Monte Carlo methods for economics and finance.

Econometric Reviews, *31*(3), 245–296.

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

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*(3), 197–208.

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.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Gallant, A. R., & Tauchen, G. (1998). Reprojecting partially observed systems with application to interest rate diffusions. *Journal of the American Statistical Association*, *93*(441), 10–24.

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).

Gouriéroux, C., & Monfort, A. (1996). *Simulation-based econometric methods.* Oxford University Press.

Gourieroux, C., Monfort, A., & Renault, E. (1993). Indirect inference. *Journal of Applied Econometrics*, *8*(S1), S85–S118.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica*, *50*(4), 1029–1054.

Harvey, A. C., Ruiz, E., & Shephard, N. (1994). Multivariate stochastic variance models. *The Review of Economic Studies*, *61*(2), 247–264.

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

Julier, S. J., & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition vi* (Vol. 3068, pp. 182–193).

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, *82*(Series D), 35–45.

Kitagawa, G. (1987). Non-Gaussian state space modeling of nonstationary time series. *Journal of the American Statistical Association*, *82*(400), 1032–1041.

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.

Kolm, P. N., & Maclin, L. (2010). Algorithmic trading. *Encyclopedia of Quantitative Finance*.

Kolmogorov, A. N. (1941). Stationary sequences in Hilbert space. *Bull. Math. Univ. Moscow*, *2*(6), 1–40.

London Stock Exchange Group. (2025). *LSEG Workspace.* Software platform. (`https://www.lseg.com/en/data-analytics/products/workspace/`)

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, *14*(1), 113–147.

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, *57*(5), 995–1026.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(6), 983–999.

Peng, W., Coleman, T., & Mentch, L. (2022). Rates of convergence for random forests via generalized u-statistics. *Electronic Journal of Statistics*, *16*(1), 232–292.

Pitt, M. K., & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, *94*(446), 590–599.

Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, *7*(2), 186–199.

Raynal, L., Marin, J.-M., Pudlo, P., Ribatet, M., Robert, C. P., & Estoup, A. (2019). ABC random forests for Bayesian parameter inference. *Bioinformatics*, *35*(10), 1720–1728.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st acm conference on electronic commerce* (pp. 158–166).

Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, *136*(12), 4629–4640.

Snyder, C., Bengtsson, T., & Morzfeld, M. (2015). Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, *143*(11), 4750–4761.

Sorenson, H. W., & Alspach, D. L. (1971). Recursive Bayesian estimation using Gaussian sums. *Automatica*, *7*(4), 465–479.

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

van de Geer, S. A. (2000). *Empirical processes in M-estimation* (Vol. 6). Cambridge University Press.

Wiener, N. (1949). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications* (Vol. 113) (No. 21). MIT press Cambridge, MA.

Zhang, T., & Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, *33*(4), 1538–1579.

Appendix

A Covariates for Fixed-Interval Smoothing

In the following we describe an approach to determine the covariates $\mathcal{C}_t = y_{s:n}$ for fixed-interval smoothing, consisting of the W observations closed to time t. A centered window of size W can have at most $h = \lfloor W/2 \rfloor$ observations on either side of time t. However, if t is near the endpoints of the time range, the window will be asymmetric. Hence, if $t + h > T$, we maximize the number of observations on the right-hand side by setting $n = T$. The lower index is then set as $s = \max\{n - W + 1, 1\}$ to establish a window of size W. Conversely, if $t + h \leq T$, the lower index $s = \max\{t - h, 1\}$ maximizes the number of observations on the left side without exceeding h, with the corresponding upper index set to $n = \min\{s + W - 1, T\}$. The resulting covariate set consists of the W observations nearest to time t.

B Proofs

B.1 Proof of Theorem 1

Since the process $\{y_t\}_{t\in\mathbb{N}}$ is assumed to be strictly stationary, the same holds for the covariate sets $\mathcal{C}_t$ for $t \geq W$, as they satisfy the rolling window condition in (6). Additionally, since the Monte Carlo draws are based on a strictly stationary process, it follows that for $t \geq W$ the regression functions $\widehat{f}_t^N(\cdot)$ are identically distributed (ID) with respect to the time index t. The above implies that the filtered estimates $\widehat{x}_t^N = \widehat{f}_t^N(\mathcal{C}_t)$ are also ID over time for $t \geq W$. Moreover, since the true states $\{x_t\}_{t\in\mathbb{N}}$ are assumed to be strictly stationary and thus ID over time, we obtain the following inequality:

$$\sup_{t\geq W} \mathbb{E}\,|x_t - \widehat{x}_t^N| \leq \sup_{t\geq W}(\mathbb{E}\,|x_t| + \mathbb{E}\,|\widehat{x}_t^N|) \leq \sup_{t\geq W} \mathbb{E}\,|x_t| + \sup_{t\geq W} \mathbb{E}\,|\widehat{x}_t^N| = \mathbb{E}\,|x_W| + \mathbb{E}\,|\widehat{x}_W^N| < \infty, \tag{21}$$

where the boundedness of the moments $\mathbb{E}\,|x_W| = \mathbb{E}\,|x_1| < \infty$ and $\mathbb{E}\,|\widehat{x}_W^N| < \infty$ holds by assumption. By similar reasoning, we have

$$\sup_{1\leq t\leq W-1} \mathbb{E}\,|x_t - \widehat{x}_t^N| \leq \sup_{1\leq t\leq W-1} (\mathbb{E}\,|x_t| + \mathbb{E}\,|\widehat{x}_t^N|) = \mathbb{E}\,|x_1| + \sup_{1\leq t\leq W-1} \mathbb{E}\,|\widehat{x}_t^N| < \infty.$$

Taken together, the above implies that

$$\sup_{t\in\mathbb{N}} \mathbb{E}\,|x_t - \widehat{x}_t^N| < \infty,$$

which establishes the result.

□

B.2 Proof of Theorem 2

For conciseness, we assume (without loss of generality) that the linear Gaussian SSM has no intercepts, as was done with the linear XMC filter in (9). By the triangle inequality,

$$\mathbb{E}_y\,|x_t^* - \widehat{x}_t^N| = \mathbb{E}_y \left| f_t^*(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t^N\right)\right| \leq \mathbb{E}_y \left| f_t^*(\mathcal{F}_t) - \widehat{f}_t^N(\mathcal{F}_t)\right| + \mathbb{E}_y \left| \widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t^N\right)\right|, \tag{22}$$

where, by Assumptions A2.1 and A2.2, the optimal filtered estimates $x_t^* = \mathbb{E}[x_t|\mathcal{F}_t]$ correspond to the Kalman filter. We start by focusing on the first term, $\mathbb{E}_y\,|f_t^*(\mathcal{F}_t) - \widehat{f}_t^N(\mathcal{F}_t)|$, in which the information set is used as covariate set.

By the joint normality of x_t and $\mathcal{F}_t$, the conditional expectation is linear:

$$\mathbb{E}[x_t|\mathcal{F}_t] = \sum_{y_j\in\mathcal{F}_t} \beta_{j,t} y_j,$$

with coefficients $\beta_{j,t} \in \mathbb{R}^{N_x\times N_y}$ (Anderson & Moore, 1979, Sec. 3.1), and the corresponding prediction error v_t satisfies

$$v_t = x_t - \mathbb{E}[x_t|\mathcal{F}_t] \sim \mathrm{N}(0, \mathbb{V}\mathrm{ar}\,[x_t|\mathcal{F}_t]).$$

The above implies that the linear regression model is correctly specified:

$$x_t = \sum_{y_j \in \mathcal{F}_t} \beta_{j,t} y_j + v_t, \qquad v_t \sim \mathrm{N}(0, \mathbb{V}\mathrm{ar}\,[x_t|\mathcal{F}_t]). \tag{23}$$

Additionally, the errors v_t are independent of the covariates $y_j \in \mathcal{F}_t$ because they are jointly normal and uncorrelated, where uncorrelatedness follows from the mean independence

$$\mathbb{E}[v_t|\mathcal{F}_t] = \mathbb{E}\Big[x_t - \mathbb{E}[x_t|\mathcal{F}_t] \,\Big|\, \mathcal{F}_t\Big] = 0 = \mathbb{E}[v_t].$$

Let z_i denote the vector that arises from vertically stacking the simulated observations $y_j^{(i)}$ from the information set, so that z_i' is the i-th row of the design matrix in the least squares regression. Then, since the data used in the regressions are independent and identically distributed with respect to the index $i = 1, \ldots, N$, all standard assumptions for consistency and $\sqrt{N}$-convergence of the least squares estimator are satisfied if the matrix $\mathbb{E}[z_i z_i']$ is non-singular (e.g., Hayashi, 2000, Proposition 2.1). As the errors are independent of the covariates, the asymptotic variance matrix of the least squares estimator reduces to

$$\mathbb{V}\mathrm{ar}\,[v_t] \cdot \mathbb{E}[z_i z_i']^{-1}.$$

If $\mathbb{E}[z_i z_i']$ exists, then $\mathbb{E}[z_i z_i'] = \mathbb{V}\mathrm{ar}\,[z_i]$ given that $\mathbb{E}[z_i] = 0$. Hence, the required non-singularity of $\mathbb{E}[z_i z_i']$ follows from the assumption that the observations in $y_{1:T}$ are linearly independent. To establish the existence of $\mathbb{E}[z_i z_i']$, we consider the diagonal and off-diagonal elements separately. Assumption A2.1 implies that the second moments of the observations y_t (the diagonal elements of $\mathbb{E}[z_i z_i']$) are finite for $t = 1, \ldots, T$. For the off-diagonal elements of $\mathbb{E}[z_i z_i']$, assume without loss of generality that the observations y_j are univariate. Then, finiteness follows from Hölder's inequality as $\mathbb{E}\,|y_j y_k| \leq \sqrt{\mathbb{E}\,|y_j|^2} \cdot \sqrt{\mathbb{E}\,|y_k|^2}$ for any j, k. It follows that the least squares estimators $\widehat{\beta}_{j,t}$ in (9) are consistent for the true parameters $\beta_{j,t}$. Furthermore, they are normal with convergence rate $\sqrt{N}$:

$$\sup_t \sup_j \left|\beta_{j,t} - \widehat{\beta}_{j,t}\right| = O_P(N^{-1/2}).$$

We therefore have that as $N \to \infty$,

$$\begin{aligned}\sup_t \mathbb{E}_y \left|f_t^*(\mathcal{F}_t) - \widehat{f}_t^N(\mathcal{F}_t)\right| &= \sup_t \mathbb{E}_y \left|\sum_{y_j \in \mathcal{F}_t} \left(\beta_{j,t} y_j - \widehat{\beta}_{j,t} y_j\right)\right| \leq \sup_t \mathbb{E}_y \sum_{y_j \in \mathcal{F}_t} \left|\beta_{j,t} - \widehat{\beta}_{j,t}\right| \cdot |y_j| \\ &= \sup_t \sum_{y_j \in \mathcal{F}_t} \left|\beta_{j,t} - \widehat{\beta}_{j,t}\right| \cdot \mathbb{E}_y\,|y_j| = O_P(N^{-1/2}),\end{aligned}$$

where the second equality follows because $\beta_{j,t}$ is constant, while $\widehat{\beta}_{j,t}$ depends only on the simulated data. The final equality follows because $\mathbb{E}_y\,|y_j|$ is a (bounded) constant.

The proof is complete once it is shown that the second term on the right-hand side of (22) can be ignored, that is,

$$\sup_t \sqrt{N}\,\mathbb{E}_y \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t^N\right)\right| \xrightarrow{P} 0 \qquad \text{as} \qquad N \to \infty.$$

This follows from applying Lemma 1 with rate $r_N = \sqrt{N}$, which is valid by Assumption 1.

□

B.3 Proof of Lemma 1

Our starting point is the triangle inequality in (22), which holds for $t = 1, \ldots, T$. The first term on the right-hand side represents the error based on an XMC filter that uses the information set as covariate set. By assumption, $\mathbb{E}_y \left|f_t^*(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{F}_t\right)\right| = O_P(r_N^{-1})$. For the second term, which represents the error from use of a covariate set instead of the information set, we will show that $\mathbb{E}_y \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t^N\right)\right| = o_P(r_N^{-1})$.

For $t = 1, \ldots, T$, the following inequality holds:

$$\left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t^N\right)\right| \leq \sup_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N(\mathcal{C}_t)\right| \cdot \mathbb{I}\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\}$$

where $\mathbb{I}\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\}$ denotes the indicator function of the event $\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\}$. Consequently, for any $\epsilon > 0$, we have the following:

$$\begin{aligned}
& P\left(r_N \, \mathbb{E}_y \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t^N\right)\right| > \epsilon\right) \\
\leq & P\left(r_N \, \mathbb{E}_y \left[\sup_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t\right)\right| \cdot \mathbb{I}\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\}\right] > \epsilon\right) \\
= & P\left(r_N \, \mathbb{E}_y \left[\sup_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t\right)\right|\right] \cdot \mathbb{I}\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\} > \epsilon\right) \\
\leq & P\left(r_N \, \mathbb{E}_y \left[\sup_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} \left|\widehat{f}_t^N(\mathcal{F}_t) - \widehat{f}_t^N\left(\mathcal{C}_t\right)\right|\right] \cdot \mathbb{I}\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\} > 0\right) \\
\leq & P\left(\mathcal{C}_t^N \neq \mathcal{F}_t\right) \to 0 \quad \text{as} \quad N \to \infty.
\end{aligned}$$

The equality follows because the composition of the regularized covariate set, and hence the occurrence of the event $\left\{\mathcal{C}_t^N \neq \mathcal{F}_t\right\}$, depends only on the simulated data. The convergence step follows from Assumption 1.

It follows that the second term on the right-hand side of (22) is $o_P(r_N^{-1})$. Thus, by applying the triangle inequality in (22), we obtain:

$$|x_t^* - \widehat{x}_t^N| \leq O_P(r_N^{-1}) + o_P(r_N^{-1}) = O_P(r_N^{-1})$$

for $t = 1, \ldots, T$, which establishes the result.

□

C Regularized Covariate Set Convergence

In this section we provide sufficient conditions to guarantee that Assumption 1 holds when determining the regularized covariate set via out-of-sample optimization or penalization of the objective function. As in Section 4.2 the time series length $T \in \mathbb{N}$ is assumed to be finite, though it can be arbitrarily large.

Consider the average validation loss

$$M_N^{\mathrm{val}}(\mathcal{C}_t) = \frac{1}{N_{\mathrm{val}}} \sum_{i=1}^{N_{\mathrm{val}}} L\left(x_t^{\langle i \rangle} - \widehat{f}_t^N \left(\mathcal{C}_t^{\langle i \rangle} \right) \right), \tag{24}$$

where the superscript $i = 1, \ldots, N_{\mathrm{val}}$ indicates cases from a separate validation sample, and $N_{\mathrm{val}} = [c_{\mathrm{val}} N]$ for some $c_{\mathrm{val}} \in (0,1)$, say, $c_{\mathrm{val}} = 0.1$. The out-of-sample optimization procedure is then defined by

$$\mathcal{C}_t^N \in \operatorname*{arg\,min}_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} M_N^{\mathrm{val}}(\mathcal{C}_t). \tag{25}$$

For the penalized estimator, let $M_N^{\mathrm{tr}}(\mathcal{C}_t)$ denote the average training loss defined by analogy to (24). Then the penalization procedure we consider is given by

$$\mathcal{C}_t^N \in \operatorname*{arg\,min}_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} M_N^{\mathrm{tr}}(\mathcal{C}_t) + \pi_N(|\mathcal{C}_t|), \tag{26}$$

with penalty term $\pi_N(\cdot)$ that is increasing in the size of the covariate set, $|\mathcal{C}_t|$,

$$\pi_N(k+1) \geq \pi_N(k) \geq 0, \qquad k = 0, \ldots, T-1,$$

and vanishing with N,

$$\lim_{N \to \infty} \sup_k \pi_N(k) = 0.$$

Examples of such penalization functions are $\pi_N(|\mathcal{C}_t|) = c|\mathcal{C}_t|/\sqrt{N}$ for $c > 0$, as well as the unregularized case $\pi_N = 0$.

The regularized covariate sets defined by (25) and (26) are M-estimators, which means that consistency can be established by showing that the average loss converges uniformly over the feasible covariate sets to the deterministic limit function

$$M_\infty(\mathcal{C}_t) = \operatorname{plim}_{N\to\infty} M_N^{\text{val}}(\mathcal{C}_t) \qquad \forall\, \mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t), \tag{27}$$

and that $\mathcal{F}_t$ is "well separated" from the other feasible covariate sets (e.g., van der Vaart, 2000, Sec. 5.2). For both conditions it is helpful to note that as the time series length is finite, the same holds for the number of feasible covariate sets. Well-separatedness then reduces to the information set $\mathcal{F}_t$ being the unique minimizer of the limit objective function M_∞. Moreover, it can be shown that *pointwise* convergence of the average loss to $M_\infty(\mathcal{C}_t)$ over the covariate sets $\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)$ implies the required *uniform* convergence.

The above ideas are used to establish the following result on the convergence of the regularized covariate sets.

Proposition 1 (Convergence of regularized covariate set)**.** *Suppose the time series $y_{1:T}$ is of finite length, and let $\mathcal{F}_t$ be the unique minimizer of the limit objective function M_∞ defined in* (27)*:*

$$M_\infty(\mathcal{F}_t) < M_\infty(\mathcal{C}_t) \qquad \forall\, \mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t) \setminus \{\mathcal{F}_t\}. \tag{28}$$

Then

$$\operatorname{plim}_{N\to\infty} \mathcal{C}_t^N = \mathcal{F}_t \tag{29}$$

is implied by either of the following conditions:

(a) The regularized covariate set is defined by the out-of-sample optimization procedure in (25) *and it holds that as $N \to \infty$,*

$$\left| M_N^{\text{val}}(\mathcal{C}_t) - M_\infty(\mathcal{C}_t) \right| \xrightarrow{P} 0 \qquad \forall\, \mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t). \tag{30}$$

(b) The regularized covariate set is defined by the penalization procedure in (26) *and it*

holds that as $N \to \infty$,

$$\left|M_N^{\mathrm{tr}}\big(\mathcal{C}_t\big) - M_\infty\big(\mathcal{C}_t\big)\right| \xrightarrow{P} 0 \qquad \forall\, \mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t).$$

Proof. For Part (a), we note that because the time series $y_{1:T}$ is of finite length, there are also finitely many feasible covariate sets: $|\mathcal{P}(\mathcal{F}_t)| < \infty$. Let $J \in \mathbb{N}$ denote the number of feasible covariate sets: $\mathcal{C}_t^{[j]}$, $j = 1, \dots, J$, and let $d_j = \left|M_N^{\mathrm{val}}\big(\mathcal{C}_t^{[j]}\big) - M_\infty\big(\mathcal{C}_t^{[j]}\big)\right|$. By the convergence in (30), it holds that for any two $\epsilon, \epsilon_P > 0$, there exists $N_j \in \mathbb{N}$ such that

$$P\left(d_j > \epsilon\right) < \epsilon_P \qquad \forall\, N \geq N_j.$$

It therefore holds for all $N \geq \sup_j N_j$ that

$$\begin{aligned} P\left(\sup_{\mathcal{C}_t \in \mathcal{P}(\mathcal{F}_t)} |M_N^{\mathrm{val}}(\mathcal{C}_t) - M_\infty(\mathcal{C}_t)| > \epsilon\right) &= P\left(\sup_j d_j > \epsilon\right) = P\left(\bigcup_{j=1}^{J} \{d_j > \epsilon\}\right) \\ &\leq \sum_{j=1}^{J} P\left(d_j > \epsilon\right) < J\epsilon_P, \end{aligned}$$

where the first inequality follows by σ-subadditivity of P. The above result shows that $M_N^{\mathrm{val}}(\mathcal{C}_t)$ converges uniformly in probability to $M_\infty(\mathcal{C}_t)$ over $\mathcal{P}(\mathcal{F}_t)$, and the same holds for the objective function in Part (b) because $\pi_N(|\mathcal{C}_t|)$ was assumed to converge to zero uniformly over $\mathcal{P}(\mathcal{F}_t)$ as $N \to \infty$. Furthermore, the minimizer $\mathcal{F}_t$ is well-separated because $|\mathcal{P}(\mathcal{F}_t)| < \infty$. Therefore, Theorem 5.7 in van der Vaart (2000) applies, which guarantees the convergence in (29) for the M-estimators defined by (25) and (26).

□

The assumption in (28) means that none of the observations can be omitted without negatively impacting the predictive performance, as expressed by the mean loss. This will generally be the case when the states $\{x_t\}$ follow an autoregressive process, which holds for many, if not most SSMs used in practice. With this in mind, the above result implies that convergence of the regularized covariate sets can often be formally shown by demonstrating that (30) holds for $t = 1, \dots, T$, that is, if the average loss for specific covariate sets converges in probability to the mean loss. The following corollary provides sufficient conditions for this to hold in the case of the linear XMC filter.

Corollary 2 (Regularized covariate sets convergence for linear XMC filter)**.** *Suppose the assumptions from Theorem 2 apply (apart from Assumption 1) and the least squares estimates $\widehat{\beta}_{j,t}$ from the linear XMC filter in* (9) *take values in a bounded parameter space Ψ_t. Assume the condition in* (28) *holds for $t = 1, \dots, T$ with limit objective function*

$$M_\infty(\mathcal{C}_t) = \mathbb{E}\left(x_t - \sum_{y_j \in \mathcal{C}_t} \beta_{j,t} y_j\right)^2,$$

where the $\beta_{j,t} = \beta_{j,t}(\mathcal{C}_t) \in \Psi_t$ denote the coefficients which minimize the mean squared error criterion above. Then, the convergence in (29) *holds for $t = 1, \dots, T$ when $\mathcal{C}_t^N$ is defined via* (25) *or* (26)*.*

Proof. The proof is immediate once we establish that (30) holds for $t = 1, \dots, T$. This can be done by noting that the summands of $M_N^{\text{val}}(\mathcal{C}_t)$ are continuous in the parameters and bounded by an integrable function, which implies that they belong to the class of Glivenko-Cantelli functions (e.g., van der Vaart, 2000, p. 46). The summands are given by

$$\begin{aligned}
L\Big(x_t - \widehat{f}_t^N(\mathcal{C}_t)\Big) &= \Big(x_t - \sum_{y_j \in \mathcal{C}_t} \widehat{\beta}_{j,t} y_j\Big)^2 \\
&= x_t^2 - 2x_t \sum_{y_j \in \mathcal{C}_t} \widehat{\beta}_{j,t} y_j + \Big(\sum_{y_j \in \mathcal{C}_t} \widehat{\beta}_{j,t} y_j\Big)^2 \\
&\leq x_t^2 + 2|x_t| \sum_{y_j \in \mathcal{C}_t} |\widehat{\beta}_{j,t}| \cdot |y_j| + \Big(\sum_{y_j \in \mathcal{C}_t} |\widehat{\beta}_{j,t}| \cdot |y_j|\Big)^2 \\
&\leq x_t^2 + 2|x_t| \sum_{y_j \in \mathcal{C}_t} |\beta_{j,t}^*| \cdot |y_j| + \Big(\sum_{y_j \in \mathcal{C}_t} |\beta_{j,t}^*| \cdot |y_j|\Big)^2,
\end{aligned}$$

where $\beta_{j,t}^*$ denotes the vector of elementwise maxima of the absolute values that $\beta_{j,t}$ can take on in the bounded parameter space Ψ_t. Additionally, Assumption A 2.1 implies that the final upper bound is integrable, which implies that (30) holds for $t = 1, \ldots, T$, and the same argument applies to $M_N^{\mathrm{tr}}(\mathcal{C}_t)$.

□

In the out-of-sample optimization procedure from Section 3, the optimization in (25) is in terms of a window size W, which determines the covariate sets (e.g., via (5) for filtering). In addition, the optimization is performed only at a single time point t^*, for computational considerations. By choosing t^* as the index of the largest information set, the convergence in (29) at $t = t^*$, combined with the use of a window size parameter, implies that the convergence applies at all times $t = 1, \ldots, T$. This supports the choice of $t^* = T$ for filtering and forecasting.

D Additional Applications

This section presents several additional applications based on the local level model in (3) and the Nile data. The static parameters are set to the maximum likelihood estimates $\sigma_x = 38.329$ and $\sigma_y = 122.877$, with $\mu_1 = 0$ and $\sigma_1^2 = 10^7$ for approximate diffuse initialization. For more information on this application, see Durbin and Koopman (2012, Ch. 2).

D.1 Forecasting

The use of regression accommodates prediction based on other information sets than the one used for filtering. For example, Figure 5 shows the 1-period forecasts of the linear XMC filter ($N = 10^4$), which coincide with the ones based on the Kalman filter. At $t = 1$, the prediction is unconditional, resulting in the value $\mu_1 = 0$, while for $t > 1$ the forecasts equal the lagged predictions from filtering, $\mathbb{E}[x_{t-1}|y_{1:t-1}]$.

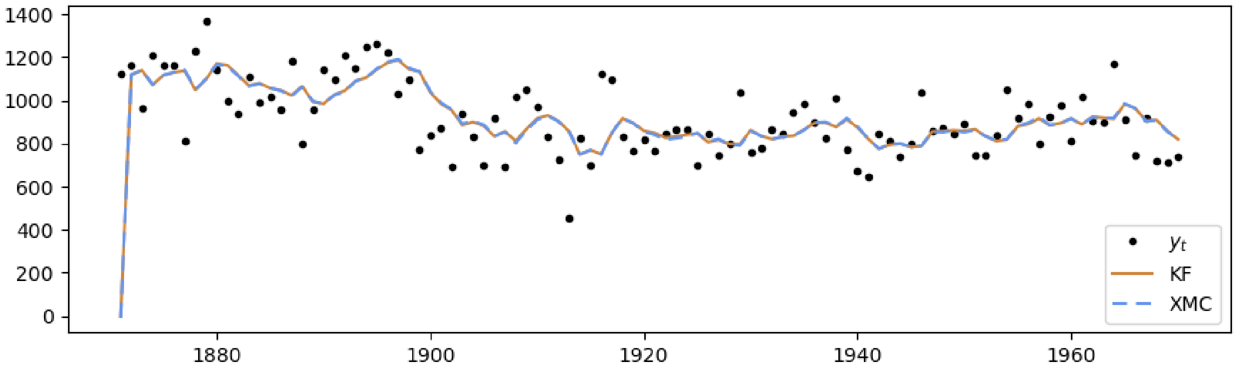


Figure 5: Forecasting analysis of the Nile data based on the local level model in (3): 1-period forecasts of the state from the Kalman filter (KF) and linear XMC filter with $N = 10^4$.

D.2 Missing Data

In practice it often occurs that some of the data are missing. The XMC method handles this issue naturally by omitting the corresponding covariates from the regressions. As illustration, we consider the local level model example from the introduction and treat the Nile measurements at times $t = 21, \ldots, 40$ and $t = 61, \ldots, 80$ as missing (Durbin & Koopman, 2012, Ch. 2). The resulting data set is shown in Figure 6. To deal with these longer sequences of missing data, the window size was set to 40. Figure 6 shows the filtered states from the linear XMC filter with $N = 10^5$ paths. The predictions are seen to coincide with those of the Kalman filter, which has an exact treatment of missing data (Durbin & Koopman, 2012, Ch. 4.10).

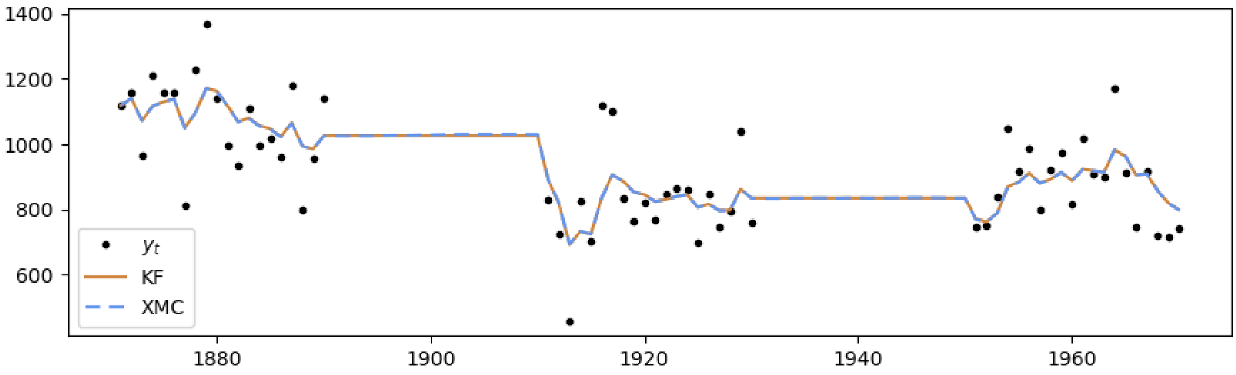


Figure 6: Filtering analysis based on the local level model in (3) and the partial Nile data set, in which the observations at time points $21, \ldots, 40$ and $61, \ldots, 80$ are treated as missing: filtered states from the Kalman filter (KF) and linear XMC filter with $N = 10^5$.

D.3 Covariate Set Regularization

To investigate how the performance of the XMC filter is impacted by the window size, we performed a simulation study using the local level model in (3) with $T = 100$. We focus on the accuracy of the filtered state at the last time point as a function of the window size, or equivalently, of the lower covariate set index $s = T - W + 1$, with the covariate set $\mathcal{C}_T$ defined via (5). In particular, the RMSE for predicting x_T was computed for the linear XMC filter with $N \in \{10^3, 10^4\}$ based on a test sample of 10^5 paths and ten repetitions of Algorithm 1 for different seeds.

Figure 7 shows the results of the simulation study. As expected, the RMSE decreases with N. For fixed N, the RMSE is non-monotonic in the lower index. Adding more observations as covariates initially improves the performance, but after some point the increase in variance from having to estimate more parameters outweighs the decrease in the bias with respect to $\mathbb{E}[x_T|y_{1:T}]$. Regarding the bias, we note that there are clear diminishing returns to adding covariates because the observations are dependent and decreasingly informative the more remote they are from the state. The optimal window size increases with N, indicating that an increase in the complexity of the regression method is warranted once more data are available. For comparison, the RMSE is also shown for the Kalman filter, which computes $\mathbb{E}[x_T|y_{1:T}]$ exactly. For $N = 10^4$, the performance of the linear XMC filter with $s = 87$ ($W = 14$) is almost indistinguishable from that of the optimal filter. The

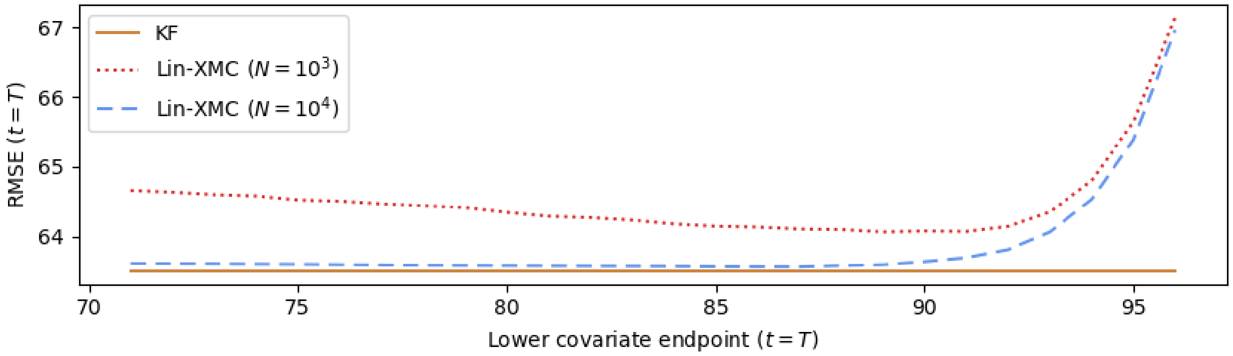


Figure 7: Root mean squared error (RMSE) for predicting the final state x_T from the local level model in (3) based on the predictions for 10^5 simulated test paths. The results are shown for the Kalman filter (KF) and linear XMC filter with $N \in \{10^3, 10^4\}$ for various values of the lower covariate set index s, and the upper index set to $T = 100$.

RMSE increases if the lower index is altered, highlighting the importance of regularization of the covariate sets.

E Bootstrap Approach for Incorporating Uncertainty in the Static Parameters

In the XMC method, the static parameters are assumed to be given, consistent with a more traditional approach to signal extraction (Chopin & Papaspiliopoulos, 2020, p. 13). In this approach, the uncertainty in these estimates is typically not considered. However, this could be incorporated using a bootstrap approach as follows.

First, generate B bootstrap samples of the observations, and use each sample to estimate the static parameters. Next, apply the XMC method to each sample for estimating the unobserved states. This results in a bootstrap distribution of state estimates, which could subsequently be used to estimate the bias, variance, and other statistical properties of the combined estimation procedure.

References

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

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

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

Hayashi, F. (2000). *Econometrics.* Princeton University Press.

van der Vaart, A. W. (2000). *Asymptotic Statistics* (Vol. 3). Cambridge University Press.