Abstract
An efficient particle Markov chain Monte Carlo methodology is proposed for the rolling-window estimation of state space models. The particles are updated to approximate the long sequence of posterior distributions as we move the estimation window. To overcome the well-known weight degeneracy problem that causes the poor approximation, we introduce a practical double-block sampler with the conditional sequential Monte Carlo update where we choose one lineage from multiple candidates for the set of current state variables. Our proposed sampler is justified in the augmented space through theoretical discussions. In the illustrative examples, it is shown to be successful to accurately estimate the posterior distributions of the model parameters.

Keywords
Double-block sampler · Forward and backward sampling · Importance sampling · Particle Gibbs · Particle Markov chain Monte Carlo · Particle simulation smoother · Rolling-window estimation · Sequential Monte Carlo · State space model · Structural change

1 Introduction
State space models have been popular and widely used in the analysis of economic and financial time series. These models are flexible and capture the dynamics of the complex economic structure. However, several structural changes have been noted in long-term economic series. If the precise time of a structural change is known, we could divide the sample period into two periods, before and after the structural change. However, this time point is usually unknown, and the change may occur gradually from one state to another. Among many alternatives proposed in the literature, one simple and popular approach to capture such structural changes is the rolling-window analysis.
In this analysis, we fix the width of the sample period, which is called “window”, and repeat estimating the unknown parameters while moving this window to the forward direction one by one. As an output of this method, we can obtain the trajectory of the estimated parameter values, and from this result, we can clearly see how the changes in the economic or financial structures occur.

As the rolling-window analysis requires running estimations many times based on different data windows, the computational efficiency is crucially important. In non-linear or non-Gaussian state space models, the likelihood is often not obtained analytically, and the maximum likelihood estimation is difficult to implement. The Markov chain Monte Carlo (MCMC) method is a popular and powerful technique used to estimate model parameters and state variables by generating random samples from the posterior distribution given a set of observed data for various complex state space models. However, in rolling estimation, simply applying the MCMC method would be too time-consuming given the need to estimate a long sequence of posterior distributions.

To overcome this difficulty, we take an alternative approach based on the sequential Monte Carlo (SMC) sampler discussed in Del Moral et al. (2006). This is effective because, in the rolling-window estimation, we can utilize the weighted samples from one posterior distribution to approximate the next posterior distribution instead of reiterating the same MCMC algorithm with the slightly different dataset. The particles consist of realized values of state variables and static parameters, which are updated when including a new observation and excluding the old observation. As we shall show in the illustrative examples of Sect. 4, a simple rolling-window sampler that is derived in a straightforward manner from the previous literature leads to the severe weight degeneracy problem, suggesting that the updating step should be constructed carefully. To fix this problem, we adopt the idea of block sampling (e.g., Doucet et al. (2006), Polson et al. (2008)), in which state variables at multiple time points are updated simultaneously when learning new information. It is highly efficient in the sense that it substantially increases the effective sample size. Based on this idea, we propose the novel sampling method, called the double-block sampler, where we sample a block of state variables when both including and excluding the information.

However, unless the time series model has a relatively simple form, finding an appropriate proposal distribution for these update steps may be difficult. Hence, instead of generating only one candidate from the proposal distribution, we generate multiple candidates and choose one of them using the conditional SMC of the particle MCMC Andrieu et al. (2010). This nested structure is similar to that of SMC$^2$ Chopin et al. (2013), Fulop and Li (2013) and nested SMC Naesseth et al. (2015), but our proposed algorithm differs in that it is derived from the particle Gibbs instead of the particle MH (Metropolis–Hastings) algorithm. As a special case of our new method, our proposed double-block sampler can be used to implement the ordinary sequential analysis by keeping all past observations. It contrasts with SMC$^2$ in that it originates from different types of the particle MCMC algorithms.

The remainder of the paper is organized as follows. In Sect. 2, we introduce the simple rolling-window sampler for state space models and point out that such a sampler derived from the conventional filtering algorithm causes the serious weight degeneracy phenomenon. Section 3 introduces a double-block sampler to overcome this difficulty.
Section 4 provides illustrative examples and, in Sect. 5, theoretical justifications of the proposed method are provided. Section 6 concludes the paper.

2 Particle rolling MCMC in general state space models

2.1 Rolling-window estimation in general state space model

Consider the state space model which consists of a measurement equation, a state equation with an observation vector \( y_t \), and an unobserved state vector \( x_t \) given a static parameter vector \( \theta \). For the prior distribution of \( \theta \), we let \( p(\theta) \) denote its prior probability density function. Further define \( x_{s:t} \equiv (x_s, x_{s+1}, \ldots, x_t) \) and \( y_{s:t} \equiv (y_s, y_{s+1}, \ldots, y_t) \). We assume that the distribution of \( y_t \) given \((y_{1:t-1}, x_{1:t}, \theta)\) depends exclusively on \( x_t \) and \( \theta \) and that the distribution of \( x_t \) given \((x_{1:t-1}, \theta)\) depends only on \( x_{t-1} \) and \( \theta \). The corresponding probability density functions are noted as follows:

\[
p(y_t | x_{1:t-1}, \theta) = p(y_t | x_t, \theta) \equiv g_\theta(y_t | x_t), \quad t = 1, \ldots, n, \tag{1}
\]

\[
p(x_t | x_{1:t-1}, \theta) = p(x_t | x_{t-1}, \theta) \equiv f_\theta(x_t | x_{t-1}), \quad t = 2, \ldots, n, \tag{2}
\]

where \( p(x_1 | \theta) \equiv \mu_\theta(x_1) \) denotes the density of the marginal distribution given \( \theta \), or the stationary distribution if exists.

We also incorporate the correlation between \( y_t \) and \( x_{t+1} \), which is conditional on \( x_t \) since we consider such an example, the realized stochastic volatility (RSV) model, for the financial time series (see, e.g., a seminal work by Takahashi et al. (2009)) in our illustrative example. It is a stochastic volatility model with an additional measurement equation for the realized volatility. Let \( y_t = (y_{1,t}, y_{2,t})' \) where \( y_{1,t} \) and \( y_{2,t} \) denote the daily log return and the logarithm of the realized volatility (variance) at time \( t \). Let \( x_t \) denote the latent log volatility which is assumed to follow the stationary AR(1) process. The RSV model is defined as follows:

\[
y_{1,t} = \exp(x_t/2)\epsilon_t, \quad t = 1, \ldots, T \tag{3}
\]

\[
y_{2,t} = x_t + \xi + u_t, \quad t = 1, \ldots, T \tag{4}
\]

\[
x_{t+1} = \mu + \phi(x_t - \mu) + \eta_t, \quad t = 1, \ldots, T, \tag{5}
\]

\[
x_1 = \mu + \frac{1}{\sqrt{1 - \phi^2}}\eta_0, \quad \eta_0 \sim \mathcal{N}(0, \sigma^2_\eta), \quad |\phi| < 1, \tag{6}
\]

where

\[
\begin{pmatrix}
\epsilon_t \\
u_t \\
\eta_t
\end{pmatrix} \sim \mathcal{N}
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
, \quad
\begin{pmatrix}
1 & 0 & \rho\sigma_\eta \\
0 & \sigma^2_u & 0 \\
\rho\sigma_\eta & 0 & \sigma^2_\eta
\end{pmatrix}
, \tag{7}
\]

\( \mathcal{N}(\mu, \Sigma) \) denotes a normal distribution with mean \( \mu \) and covariance matrix \( \Sigma \), and \( \theta = (\mu, \phi, \sigma^2_\eta, \xi, \sigma^2_u, \rho)' \) is the static parameter vector. The correlation \( \rho \) between \( \epsilon_t \) and \( \eta_t \) is introduced to express the leverage effect. The effect is often negative in
empirical studies, which implies that the decrease in the today’s log return is followed by the increase in the log volatility on the next day (e.g., Omori et al. 2007). In this case, we express the dependence of \( y_t \) on \( x_{t+1} \) (or \( x_{t+1} \) on \( y_t \)) as follows:

\[
p(y_t \mid x_{1:t} + 1, y_{1:t-1}, \theta) = p(y_t \mid x_t, x_{t+1}, \theta) \equiv g_\theta(y_t \mid x_t, x_{t+1}), \quad t = 1, \ldots, n, \tag{8}
\]

\[
p(x_{t+1} \mid x_{1:t}, y_{1:t}, \theta) = p(x_{t+1} \mid x_t, y_t, \theta) \equiv f_\theta(x_{t+1} \mid x_t, y_t), \quad t = 1, \ldots, n - 1. \tag{9}
\]

In the rolling-window estimation of time series, the number of observations (or the window size) in the sample period is fixed. When the fixed window size is \( L + 1 \), we repeat estimating the posterior distributions of \( \theta \) and \( x_{t-L:t} \) given the data \( y_{t-L:t} \) for \( t = L + 1, L + 2, \ldots \). To simplify the notation, we define \( s := t - L \). Then, the posterior of interest at time \( t \) is written as

\[
\pi(x_{s:t}, \theta \mid y_{s:t}) \propto p(\theta) \mu_\theta(x_s) g_\theta(y_s \mid x_s) \prod_{j=s+1}^{t} f_\theta(x_j \mid x_{j-1}, y_{j-1}) g_\theta(y_j \mid x_j), \tag{10}
\]

or, equivalently,

\[
\pi(x_{s:t}, \theta \mid y_{s:t}) \propto p(\theta) \mu_\theta(x_s) \prod_{j=s+1}^{t} f_\theta(x_j \mid x_{j-1}, y_{j-1}) g_\theta(y_{j-1} \mid x_{j-1}, x_j) g_\theta(y_j \mid x_j). \tag{11}
\]

### 2.2 Simple rolling-window sampler

We first describe a simple rolling-window sampler that is derived in a straightforward manner from the previous literature. The estimation procedure consists of two steps: When we have the Monte Carlo sample from the posterior \( \pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) \), in Step 1, we incorporate the information of the new observation \( y_t \), that is, update the sample so that they approximate the new posterior \( \pi(x_{s-1:t}, \theta \mid y_{s-1:t}) \). In Step 2, we discard the information of \( y_{s-1} \), or sample from the posterior \( \pi(x_{s:t}, \theta \mid y_{s:t}) \). We use the SMC sampler in Del Moral et al. (2006) as detailed below.

**Step 1.** Assume that, at time \( t - 1 \), we have a collection of particles \( (x_{s-1:t-1}^n, \theta^n) \) with the importance weight \( W_{[s-1,t-1]}^n, (n = 1, \ldots, N) \) which is a discrete approximation of \( \pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) \). We include a new observation \( y_t \) in the information set and aim to sample from \( \pi(x_{s-1:t}, \theta \mid y_{s-1:t}) \). Given the current sample \( (x_{s-1:t-1}, \theta) \) from \( \pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) \), we propose a candidate \( x_t \) using some proposal density \( q_{l, \theta}(x_t \mid x_{t-1}, y_t) \). The incremental weight is given as

\[
\frac{\pi(x_{s-1:t}, \theta \mid y_{s-1:t})}{\pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) q_{l, \theta}(x_t \mid x_{t-1}, y_t)} = \frac{p(x_t, y_t \mid x_{s-1:t-1}, y_{s-1:t-1}, \theta) q_{l, \theta}(x_t \mid x_{t-1}, y_t) p(y_t \mid y_{s-1:t-1})}{q_{l, \theta}(x_t \mid x_{t-1}, y_t) g_\theta(y_t \mid x_t)} \frac{f_\theta(x_t \mid y_t, x_{t-1}) g_\theta(y_t \mid x_t)}{f_\theta(x_t \mid y_t, x_{t-1}) g_\theta(y_t \mid x_t)},
\]

1 In the discussion, the notation \([k, l] \) \((k < l)\) works as “an index” that indicates the interval from \( k \) to \( l \).
so we generate $x^n_t \sim q_t, \theta^n (x^n_t \mid x^n_{t-1}, y_t)$ and compute the importance weight

$$W^n_{[s-1,t]} \propto \frac{f^n q^n(x^n_t \mid x^n_{t-1}, y_{t-1}) g^n(y_t \mid x^n_t)}{q_t, \theta^n (x^n_t \mid x^n_{t-1}, y_t)} \times W^n_{[s-1,t-1]}, \quad (12)$$

where the weights are normalized so that $\sum_n W^n_{[s-1,t]} = 1$. We note that the importance weights introduced in the rest of the discussion are normalized as well.

**Step 2.** We exclude the old observation $y_{s-1}$ from the information to construct a Monte Carlo sample approximating $\pi(x_{s:t}, \theta \mid y_{s:t})$. To match the dimension of the previous target distribution $\pi(x_{s-1:t}, \theta \mid y_{s-1:t})$ and the new target distribution, we formulate this step as an operation to sample from

$$\pi(x_{s:t}, \theta \mid y_{s:t}) p(x_{s-1} \mid x_s, \theta),$$

where $\pi(x_{s:t}, \theta \mid y_{s:t})$ is clearly a marginal density. Then, the ratio of the two target distributions, which we use as the incremental weight, is

$$\frac{\pi(x_{s:t}, \theta \mid y_{s:t}) p(x_{s-1} \mid x_s, \theta)}{\pi(x_{s-1:t}, \theta \mid y_{s-1:t})} \propto g_{\theta}(y_{s-1} \mid x_{s-1}, x_s)^{-1},$$

where we used a decomposition of the posterior density based on the Markov structure:

$$\pi(x_{s-1:t}, \theta \mid y_{s-1:t}) \propto p(x_{s-1} \mid x_s, \theta) \pi(x_{s:t}, \theta \mid y_{s-1:t}) g_{\theta}(y_{s-1} \mid x_{s-1}, x_s).$$

In this step, we do not generate new values and update the particles by reweighting as follows:

$$W^n_{[s,t]} \propto g_{\theta^n}(y_{s-1} \mid x^n_{s-1}, x^n_s)^{-1} \times W^n_{[s-1,t]}, \quad (13)$$

The particle for $x_{s-1}$ is removed since it is no longer used.

**Resampling.** Every time we complete Step 1 or 2, we compute some degeneracy criteria such as the effective sample size (ESS),

$$\text{ESS}_{[s-1:t]} = \left\{ \frac{1}{N} \sum_{n=1}^{N} \left( W^n_{[s-1,t]} \right)^2 \right\}^{-1}, \quad (14)$$

and the particles are resampled if $\text{ESS} < cN$ (e.g., $c = 0.5$). After resampling, we additionally refresh the particles with the MCMC algorithm. We note that one can also update particles using the particle Gibbs sampler (Andrieu et al. (2010)).

**Remark** The choice of the MCMC algorithm in the resampling step is very important. If the mixing of the algorithm is poor, the updated particles still have similar values, so the posterior distributions approximated by them are no longer reliable. The computational cost of MCMC is also crucial especially if the particles are resampled many times. In practice, however, the MCMC algorithms we can use tend to be complicated.
and computationally costly unless the time series model is simple, so in the actual estimations, we want to trigger the resampling and MCMC steps as infrequently as possible. This is one of our main motivations to improve the simple algorithm to avoid the weight degeneracy problem, as we discuss in the next section.

The procedure described above is summarized in Algorithm 1.

Algorithm 1: Simple rolling-window sampler

Let \((x^n_{s-1:t-1}, \theta^n)\) denote the sample from \(\pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1})\) with the weight \(W^n_{[s-1:t-1]}\) \((n = 1, \ldots, N)\).

**Step 1:** Generate \(x^n_t \sim q_{t, \theta^n}(x^n_t \mid x^n_{t-1}, y_t)\) and set

\[
W^n_{[s-1:t-1]} \propto \frac{f_{\theta^n}(x^n_t \mid x^n_{t-1}, y_{t-1})g_{\theta^n}(y_t \mid x^n_t)}{q_{t, \theta^n}(x^n_t \mid x^n_{t-1}, y_t)} \times W^n_{[s-1:t-1]}.
\]

**Step 2:** Update the weight

\[
W^n_{[s,t]} \propto g_{\theta^n}(y_{s-1} \mid x^n_{s-1}, x^n_t)^{-1} \times W^n_{[s-1:t-1]},
\]

and discard \(x^n_{s-1}\).

**Resampling:** After Step 1 or 2, if ESS < \(cN\), resample the particles and update them with the MCMC algorithm.

### 2.3 Weight degeneracy problem

Using the importance weight (13) in Step 2 is obviously problematic because it would take an extremely high value when \(g_{\theta} \) is close to 0. This causes the ESS to rapidly drop and triggers the MCMC update steps many times, which makes the estimation time-consuming. Further, in Step 1, one might think it will work without any problem as long as we choose an appropriate proposal distribution \(q_{t, \theta} \). However, as we shall see in illustrative examples in Sect. 4, this step also causes a serious degeneracy problem. In Sect. 3, we overcome this difficulty of the weight degeneracy by proposing a novel sampling method called “a double-block sampler” with the conditional SMC update.

### 3 Particle rolling MCMC with double-block sampler

We consider sampling a block of state variables when we add the new observation or remove the old observation. For example, we update values of \(\{x^n_{t-K:t-1}\}_{n=1}^N\) in addition to generating \(\{x^n_{t}\}_{n=1}^N\) when we learn the information of \(y_t\). We call this process the forward block sampling (Step 1), and the backward block sampling (Step...
2) can also be defined in a similar manner. The double-block sampler addresses the weight degeneracy problem by reducing the path dependence between the new particle and the old particle values that are not updated.

### 3.1 Idealized double-block sampler

We first consider the ‘idealized’ double-block sampler where we assume an appropriate $K + 1$ dimensional proposal distribution is available for the block sampling. The SMC algorithm proceeds as the simple algorithm described in Sect. 2.2, and we move from $\pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1})$ to $\pi(x_{s-1:t}, \theta \mid y_{s-1:t})$ in Step 1, and from $\pi(x_{s-1:t}, \theta \mid y_{s-1:t})$ to $\pi(x_{s:t}, \theta \mid y_{s:t})$ in Step 2.

**Step 1** To incorporate the information of $y_t$, we generate new values from the proposal kernel $\pi(x_{t-K:t}^\dagger \mid x_{s-1:t-K-1}, y_{s-1:t}, \theta)$. It should be noted that $x_{t-K:t-1}$ and $x_{t-K:t}^\dagger$ are different variables though they are defined in the same time period. The target distribution of this step defined in the augmented space is given by

$$
\pi(x_{s-1:t-K-1}, x_{t-K:t}^\dagger, \theta \mid y_{s-1:t}) \pi(x_{t-K:t} \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta),
$$

where the marginal density $\pi(x_{s-1:t-K-1}, x_{t-K:t}^\dagger, \theta \mid y_{s-1:t})$ is of our main interest. Technically speaking, we use $\pi(x_{t-K:t} \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)$ as the backward kernel in the general framework provided in Del Moral et al. (2006). Since this distribution is rewritten as

$$
\pi(x_{s-1:t-K-1}, x_{t-K:t}^\dagger, \theta \mid y_{s-1:t}) = \frac{p(x_{s-1:t-K-1}, x_{t-K:t}^\dagger, y_t, \theta \mid y_{s-1:t-1})}{p(y_t \mid y_{s-1:t-1})} = \frac{\pi(x_{s-1:t-K-1}, \theta \mid y_{s-1:t-1})}{p(y_t \mid y_{s-1:t-1})} \times \frac{p(x_{t-K:t}^\dagger, y_t \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)}{p(y_t \mid y_{s-1:t-1})} \times \frac{p(y_t \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)}{p(y_t \mid y_{s-1:t-1})},
$$

the unnormalized incremental weight is

$$
\frac{\pi(x_{s-1:t-K-1}, x_{t-K:t}^\dagger, \theta \mid y_{s-1:t}) \pi(x_{t-K:t} \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)}{\pi(x_{s-1:t-K-1}, \theta \mid y_{s-1:t-1}) \pi(x_{t-K:t}^\dagger \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)} = \frac{p(y_t \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)}{p(y_t \mid y_{s-1:t-1})} \times \frac{p(y_t \mid x_{s-1:t-K-1}, y_{s-1:t-1}, \theta)}{p(y_t \mid y_{s-1:t-1})}.
$$
Thus, in the SMC algorithm, we generate proposal values from \( \pi(x_{t-K:t-1}^n | x_{s-1:t-K-1}, y_{s-1:t}, \theta) \) and update the importance weight as
\[
W_{[s-1:t]}^n \propto p(y_t | x_{t-K:t-1}^n, y_{s-1:t-1}, \theta^n) \times W_{[s-1:t-1]}^n,
\]
noting that \( p(y_t | x_{s-1:t-K-1}, y_{s-1:t-1}, \theta) = p(y_t | x_{t-K-1}, y_{s-1:t-1}, \theta) \).

**Step 2.** We remove the old observation \( y_{s-1} \) from the information set and to this end, we update the latent states close to the time \( s-1 \), especially \( x_{s-1:t} \), from the proposal density \( \pi(x_{s:s+K-1}^\uparrow | x_{s+K:t}, y_{s:t}, \theta) \). Then, the target distribution is written as
\[
\pi(x_{s:s+K-1}^\uparrow, x_{s+K:t}, \theta | y_{s:t}) \pi(x_{s-1:s+K-1} | x_{s+K:t}, y_{s-1:t}, \theta),
\]
where we are interested in the marginal density \( \pi(x_{s:s+K-1}^\uparrow, x_{s+K:t}, \theta | y_{s:t}) \). In the terminology of Del Moral et al. (2006), we use \( \pi(x_{s-1:s+K-1} | x_{s+K:t}, y_{s-1:t}, \theta) \) as the backward kernel. The unnormalized incremental weight is
\[
\frac{\pi(x_{s:s+K-1}^\uparrow, x_{s+K:t}, \theta | y_{s:t}) \pi(x_{s-1:s+K-1} | x_{s+K:t}, y_{s-1:t}, \theta)}{\pi(x_{s-1:t}, \theta | y_{s-1:t}) \pi(x_{s:s+K-1}^\uparrow | x_{s+K:t}, y_{s:t}, \theta)} = \frac{p(y_{s-1} | x_{s+K:t}, y_{s:t}, \theta)}{p(y_{s-1} | x_{s+K:t}, y_{s:t}, \theta)},
\]
since
\[
\pi(x_{s:s+K-1}^\uparrow, x_{s+K:t}, \theta | y_{s:t}) = \pi(x_{s:s+K-1}^\uparrow | x_{s+K:t}, y_{s:t}, \theta) \pi(x_{s+K:t}, \theta | y_{s:t})
\]
and
\[
\pi(x_{s-1:t}, \theta | y_{s-1:t}) = \pi(x_{s-1:s+K-1} | x_{s+K:t}, y_{s-1:t}, \theta) \pi(x_{s+K:t}, \theta | y_{s-1:t})
\]
\[
= \pi(x_{s-1:s+K-1} | x_{s+K:t}, y_{s-1:t}, \theta) \times \frac{p(x_{s+K:t}, y_{s-1}, \theta | y_{s:t})}{p(y_{s-1} | y_{s:t})} \times \frac{p(y_{s-1} | x_{s+K:t}, y_{s:t}, \theta)}{p(y_{s-1} | y_{s:t})}.
\]
Noting that \( p(y_{s-1} | x_{s+K:t}, y_{s:t}, \theta) = p(y_{s-1} | x_{s+K}, y_{s:t}, \theta) \), we set \( x_{s:t} = (x_{s:s+K-1}^\uparrow, x_{s+K:t}) \) and update the importance weight
\[
W_{[s:t]}^n \propto p(y_{s-1} | x_{s+K}^n, y_{s:t}, \theta^n)^{-1} \times W_{[s-1:t]}^n,
\]
and discard \( x_{s-1}^n \).

The sampling algorithm is summarized in Algorithm 2. Recall that as in Algorithm 1, we check the ESS after Step 1 or 2 and run the resampling step if necessary.

### 3.2 Practical double-block sampler

In practice, it is often difficult to find an ‘idealized’ proposal distribution for the block sampling. Hence, we adopt the approach of the conditional SMC update for the particle
Algorithm 2: Idealized double-block sampler

Let \( (x_{s-1:t-1}^n, \theta^n) \) denote the sample from \( \pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) \) with the weight \( W_{n[s-1,t-1]} \) \((n = 1, \ldots, N)\).

**Step 1:** Generate \( x_{t-K:t}^n \sim \pi(x_{t-K:t}^n \mid x_{s-1:t-K-1}^n, y_{s-1:t-1}, \theta^n) \) \((s - 1 \leq t - K - 1)\) and set

\[
W_{n[s-1,t-1]}^n \propto p(y_t \mid x_{t-K-1}^n, y_{s-1:t-1}, \theta^n) \times W_{n[s-1,t-1]}.
\]

**Step 2:** Generate \( x_{s:s+K-1}^n \sim \pi(x_{s:s+K-1}^n \mid x_{s+K:t}^n, y_{s:t}, \theta^n) \) and set

\[
W_{n[s,t]}^n \propto p(y_{s-1} \mid x_{s+K}, y_{s:t}, \theta^n)^{-1} \times W_{n[s-1,t]}.
\]

and discard \( x_{n-1}^n \).

**Resampling:** After Step 1 or 2, if ESS \( < cN \), resample the particles and update them with the MCMC algorithm.

Gibbs sampler Andrieu et al. (2010), which considers the artificial target density and generates a cloud of values for one particle path.

In Step 1, we have samples from the old target density \( \pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) \), and generate the indices \( k_{t-K:t-1} = (k_{t-K}, \ldots, k_{t-1}) \) and a cloud of particles from the proposal kernel \( \psi_{\theta} \) defined in (40). The new target density is \( \pi(x_{s-1:t-K-1}, x_{t-K:t}, \theta \mid y_{s-1:t}) \) with the backward kernel \( \hat{\pi} / \pi(x_{s-1:t-K-1}, x_{t-K:t}, \theta \mid y_{s-1:t}) \) where \( \hat{\pi} \) is defined in (41). We set \( x_{s-1:t} = (x_{s-1:t-K-1}, x_{t-K:t}) \) which is the sample from the new target density with the unnormalized incremental weight \( \hat{p}(y_t \mid x_{t-K-1}^n, y_{s-1:t-1}, \theta^n) \) in (26).

In Step 2, we have samples from the old target density \( \pi(x_{s-1:t}, \theta \mid y_{s-1:t}) \), and generate the indices \( k_{s-1:s+K-1} \) and a cloud of particles from the proposal kernel \( \psi_{\theta} \) defined in (45). The new target density is \( \pi(x_{s:K-1}, x_{s+K}, \theta \mid y_{s:t}) \) with the backward kernel \( \hat{\pi} / \pi(x_{s:K-1}, x_{s+K}, \theta \mid y_{s:t}) \) where \( \hat{\pi} \) defined in (46). We set \( x_{s:t} = (x_{s:K-1}, x_{s+K}) \) which is the sample from the new target density with the unnormalized incremental weight \( \hat{p}(y_{s-1} \mid x_{s+K}, y_{s:t}, \theta^n)^{-1} \) in (32). Details are given below.

---

2 The marginal density of \( \hat{\pi} \) is \( \pi(x_{s-1:t-K-1}, x_{t-K:t} \mid \theta \mid y_{s-1:t}) \) as shown in Proposition 5.1 with \( x_{t-K:t} = (x_{t-K}^\dagger, \ldots, x_{t}^\dagger) \).

3 The marginal density of \( \hat{\pi} \) is \( \pi(x_{s:K-1}, x_{s+K}, \theta \mid y_{s:t}) \) as shown in Proposition 5.3 with \( x_{s:K-1}^\dagger = (x_{s}^\dagger, \ldots, x_{s+K-1}^\dagger) \).
3.2.1 Forward block sampling (step 1)

We first generate a number of candidates \( x_{t-K:t-1}^{n,m} \) with the current values \( x_{t-K:t-1}^{n} \) fixed using the conditional SMC. Then, for each \( x_{t-K:t-1}^{n,m} \), we generate \( x_{t,K}^{n,m} \). In this ‘local particle filtering’, we resample the particles at \( t - K + 1, \ldots, t \). This operation is equivalent to choosing the ‘parent’ \( x_{j}^{n,m} \) for \( x_{j+1}^{n,m} \) \( (j = t - K, \ldots, t - 1) \).

Using this terminology, if we choose one particle \( x_{t}^{n,m} \), its ‘ancestors’ are uniquely determined from \( x_{j}^{n,m} \) \( (j = t - K, \ldots, t - 1) \). We call this descendant and its ancestors the ‘lineage’. In the conditional SMC step, fixing the current values \( x_{t-K:t-1}^{n} \) is seen as fixing one lineage by choosing their indices \( k_{j} \) \( (j = t - K, \ldots, t - 1) \) (where we drop the superscript \( n \) for simplicity) which follows the rule

\[
a_{j}^{k_{j+1}} = k_{j}, \quad j = t - K, \ldots, t - 1.
\]

In addition, the index of their descendant is determined as \( k_{t} \).

After generating \( x_{t-K:t}^{n,m} \) \( (m = 1, \ldots, M) \), we choose one lineage to store as the next values of \( x_{t-K:t}^{n} \). This is equivalent to sampling a random index \( k_{t} \) for the candidate \( x_{t}^{n,m} \) and identifying the ancestors for which indices are obtained by following the rule

\[
a_{j}^{k_{*}} = k_{j}, \quad j = t - K, \ldots, t - 1.
\]

Moreover, we can improve its efficiency by implementing ‘smoothing’ for the generated candidates following the algorithm reported in Whiteley et al. (2010). In this smoothing step, we again choose \( k_{j}^{*} \) for \( j = t - K, \ldots, t - 1 \) randomly. This manipulation of breaking the relationship between the parent and the child in the lineage is effective in improving the mixing property, or sampling values of \( x_{t-K:t-1}^{n} \) that may be different from the lineages obtained in the previous step.

The detailed algorithm is provided below. We fix one lineage in Step 1-1(a) and implement the conditional SMC in Steps 1-1(b) and 1-1(c). The candidates for \( x_{t}^{n} \) are generated in Step 1-1(d) and we compute the importance weight for the \( n \)-th particle in the ‘global particle filtering’ in Step 1-2. The smoothing is implemented in Step 1-3.

1. We generate \( x_{t-K:t}^{n} \) \( \sim \pi(x_{t-K:t}^{n} \mid x_{s-1:t-K-1}^{n}, y_{s-1:t}, \theta^{n}) \) using the conditional SMC update:
   (a) Sample \( k_{j} \) from \( \{1, \ldots, M\} \) with probability \( 1/M \) \( (j = t - K, \ldots, t - 1) \) and set
   \[
   (x_{t-K}^{n,k_{t-K}}, \ldots, x_{t-1}^{n,k_{t-1}}) = x_{t-K:t-1}^{n}, \quad (a_{t-K}^{k_{t-K}}, \ldots, a_{t-1}^{k_{t-1}}) = (k_{t-K}, \ldots, k_{t-1}),
   \]
   where \( x_{t-K:t-1}^{n} \) is a current sample with the importance weight \( W_{s-1:t-1}^{n} \).
   (b) Set \( x_{t-K}^{n,m} = x_{t-K-1}^{n,m} \) for all \( m \) according to the convention, and sample
   \[
   x_{t-K}^{n,m} \sim q_{t-K,\theta^{n}}(\cdot \mid x_{t-K-1}^{n,m}, y_{t-K}) \quad \text{for each} \quad m \in \{1, \ldots, M\} \setminus \{k_{t-K}\}. \quad \text{Let} \quad j = t - K + 1.
   \]
(c) Sample \( a_{j-1}^m \sim \mathcal{M}(V_j^{1:M} \cdot \theta) \) and \( x_{j-1}^{n,m} \sim q_{j,\theta}(\cdot \mid x_{j-1}^{n,a_{j-1}^m}, y_j) \) for each \( m \in \{1, \ldots, M\} \setminus \{k_j\} \) where \( V_j^{1:M} = (V_j^{1}, \ldots, V_j^{M}) \) and

\[
V_{j,\theta}^m = \frac{v_{j,\theta}(x_{j-1}^{n,a_{j-1}^m}, x_{j}^{n,m})}{\sum_{i=1}^{M} v_{j,\theta}(x_{j-1}^{n,a_{j-1}^i}, x_{j}^{n,i})}, \\
\quad m = 1, \ldots, M. 
\]  

(23)

(d) If \( j < t - 1 \), set \( j \leftarrow j + 1 \) and go to (c). Otherwise, sample \( x_t^{n,m} (m = 1, \ldots, M) \) and \( k_t^* \) as follows.

(i) Sample \( x_{t-1}^{n,1} \sim q_{t-1,\theta}(\cdot \mid x_{t-1}^{n,k_{t-1}}, y_t) \).

(ii) Sample \( a_{t-1}^{n,1} \sim \mathcal{M}(V_{t-1,\theta}^{1:M}) \) and \( x_{t-1}^{n,m} \sim q_{t,\theta}(\cdot \mid x_{t-1}^{n,a_{t-1}^m}, y_t) \) for each \( m \in \{2, \ldots, M\} \).

(iii) Sample \( k_{t}^* \sim \mathcal{M}(V_{t,\theta}^{1:M}) \) and obtain \( k_{t}^* (j = t - 1, \ldots, t - K) \) using (22).

2. Let \( x_{s-1:t}^n = (x_{s-1}^n, \ldots, x_{t-K+1}^n, x_{t-K}^{n,k_{t-K}^*}, \ldots, x_t^{n,k_{t}^*}) \) and compute the importance weight\(^4\)

\[
W_{[s-1:t]}^n \propto \hat{p}(y_t \mid x_{t-K+1}^n, y_{s-1:t-1}, \theta^n) \times W_{[s-1:t-1]}^n. 
\]  

(25)

\[
\hat{p}(y_t \mid x_{t-K+1}^n, y_{s-1:t-1}, \theta^n) = \frac{1}{M} \sum_{m=1}^{M} v_{t,\theta}(x_{t-1}^{n,a_{t-1}^m}, x_{t}^{n,m}), 
\]  

(26)

where \( \hat{p}(y_t \mid x_{t-K+1}^n, y_{s-1:t-1}, \theta^n) \) can be seen as the estimate of the intractable incremental weight \( p(y_t \mid x_{t-K+1}^n, y_{s-1:t-1}, \theta^n) \) in (17) for the idealized double-block sampler.

3. Implement the particle simulation smoother to sample \( (k_{t-K}^*, k_{t-K+1}^*, \ldots, k_t^*) \) jointly. Generate \( k_{j}^* \sim \mathcal{M}(\tilde{V}_{j,\theta}^{1:M}) \), \( j = t - 1, \ldots, t - K \), recursively where

\[
\tilde{V}_{j,\theta}^m = \frac{V_{j,\theta}^m f_{\theta}(x_{j+1}^{k_{j+1}^*} \mid x_{j}^{n}, y_{j+1})}{\sum_{i=1}^{M} V_{j,\theta}^i f_{\theta}(x_{j+1}^{k_{j+1}^i} \mid x_{j}^{n}, y_{j+1})}, \quad m = 1, \ldots, M. 
\]  

(27)

and set \( x_{s-1:t}^n = (x_{s-1}^n, \ldots, x_{t-K+1}^n, x_{t-K}^{n,k_{t-K}^*}, \ldots, x_t^{n,k_{t}^*}) \).

Figure 1 illustrates an example with \( K = 2, M = 4 \) and the current sample \( (x_{s-1:t-1}^n, \theta^n) \).

\(^4\) We use the notation \( \hat{p}(y_t \mid x_{t-K+1}^n, y_{s-1:t-1}, \theta^n) \) since it is an unbiased estimator of \( p(y_t \mid x_{t-K+1}^n, y_{s-1:t-1}, \theta^n) \) as we shall show in Proposition 5.2.
1. (a) Sample $k_{t-2}$ and $k_{t-1}$ from $\{1, 2, 3, 4\}$ with probability 1/4 and suppose $k_{t-2} = k_{t-1} = 1$. We set $x_{t-2}^{n,1} = x_{t-2}^n$, $x_{t-1}^{n,1} = x_{t-1}^n$ (with the red rectangle) and $(a_{t-2}^1, a_{t-1}^1) = (1, 1)$.

(b) Set $x_{t-3} = x_{t-3}^n$ for all $m$ (with the black rectangle), and sample $x_{t-2}^{n,m} \sim q_{t-2, \theta} (\cdot | x_{t-3}^n, y_{t-2})$ for each $m \in \{2, 3, 4\}$ (with the black circle).

(c) Sample $a_{t-2}^m \sim \mathcal{M}(V_{t-2,\theta})$ for $m \in \{2, 3, 4\}$ and suppose $a_{t-2}^2 = 2, a_{t-2}^3 = 3, a_{t-2}^4 = 3$. Generate $x_{t-1}^{n,m} \sim q_{t-1, \theta} (\cdot | x_{t-2}^{n,a_{t-2}^m}, y_{t-1})$ for $m \in \{2, 3, 4\}$ (with the black circle).

(d) (i) Sample $x_{t-1}^{n,1} \sim q_{t, \theta} (\cdot | x_{t-1}^{n,1}, y_t)$.

(ii) Sample $a_{t-1}^m \sim \mathcal{M}(V_{t-1,\theta})$ for $m \in \{2, 3, 4\}$ and suppose $a_{t-1}^2 = 2, a_{t-1}^3 = 1, a_{t-1}^4 = 4$. Generate $x_{t}^{n,m} \sim q_{t, \theta} (\cdot | x_{t-1}^{n,a_{t-1}^m}, y_t)$ for $m \in \{2, 3, 4\}$.

(iii) Sample $k_{t}^* \sim \mathcal{M}(V_{j,\theta})$ and suppose $k_{t}^* = 3$. Using (22), we obtain $k_{t-1}^* = k_{t-2}^* = 1$ and select $(x_{t-3}^{n,3}, x_{t-1}^{n,1}, x_{t-2}^{n,4})$ with red lines.

2. Let $x_{s-1:t}^n = (x_{s-1}^n, \ldots, x_{t-3}^n, x_{t-2}^{n,1}, x_{t-1}^{n,1}, x_{t}^{n,3})$ and compute the importance weight.

2. Implement the particle simulation smoother to sample $(k_{s-2}^*, k_{s-1}^*, k_{s}^*)$ jointly. Generate $k_{j}^* \sim \mathcal{M}(\bar{V}_{j,\theta}, j = t-1, t-2$, recursively (with dotted lines) and suppose $k_{t-1}^* = 3$ and $k_{t-2}^* = 3$. We set $x_{s-1:t}^n = (x_{s-1}^n, \ldots, x_{t-3}^n, x_{t-2}^{n,3}, x_{t-1}^{n,3}, x_{t}^{n,3})$.

**Remark 1** As the proposal density $q_{j, \theta}$, we can either use the prior density $f_{\theta}$ or more sophisticated density that incorporates the information of the likelihood $g_{\theta}$. Even if...
Algorithm 3 (Step 1): Practical double-block sampler

Let \( (x^n_{s-1:t-1}, \theta^n) \) denote the sample from \( \pi(x_{s-1:t-1}, \theta \mid y_{s-1:t-1}) \) with the weight \( W^n_{s-1:t-1} \) (\( n = 1, \ldots, N \)).

1. We generate \( x^n_{t-K:t} \sim \pi(x^n_{t-K:t} \mid x^n_{t-K-1:t-1}, y^n_{s-1:t-1}, \theta^n) \) (\( s - 1 \leq t \leq K - 1 \)) using the conditional SMC update:

   (a) Sample \( k_j \) from \([1, \ldots, M]\) with probability \( 1/M \) (\( j = t - K, \ldots, t - 1 \)) and set
   \[
   (x^n_{t-K}, \ldots, x^n_{t-1}) = (x^n_{t-K:t-1}), \quad (a^n_{t-K}, \ldots, a^n_{t-1}) = (k^n_{t-K}, \ldots, k^n_{t-1}).
   \]

   (b) Set \( x^n_{t-K-1:t-1} = x^n_{t-K-1} \) and sample \( x^n_{t-K:m} \sim q^n_{t-K, \theta^n} (\cdot \mid x^n_{t-K-1:t-1}, y^n_{t-K}), m \in [1, \ldots, M] \setminus \{k^n_{t-K}\} \).

   (c) Sample \( a^n_{j-1} \sim \mathcal{M}(V^{-1}_{j-1, \theta^n}) \) and \( x^n_{j-1,m} \sim q^n_{j-1, \theta^n} (\cdot \mid x^n_{j-1:m-1}, y^n_{j-1}), m \in [1, \ldots, M] \setminus \{j-1\} \) where \( V^{-1}_{j-1, \theta^n} \) is given by (23).

   (d) If \( j < t - 1 \), set \( j \leftarrow j + 1 \) and go to (c). Otherwise, sample \( x^n_{t,m} \) (\( m = 1, \ldots, M \)) and \( k^n_j \) as follows.

     (i) Sample \( x^n_t \sim q^n_t (\cdot \mid x^n_{t-1:k^n_j}, y^n_t) \).

     (ii) Sample \( a^n_{t,m} \sim \mathcal{M}(V^{-1}_{t-1, \theta^n}) \) and \( x^n_{t,m} \sim q^n_{t, \theta^n} (\cdot \mid x^n_{t-1:m-1}, y^n_{t}), m \in [2, \ldots, M] \).

     (iii) Sample \( k^n_j \sim \mathcal{M}(V^{-1}_{t, \theta^n}) \) and obtain \( k^n_j \) (\( j = t - 1, \ldots, t - K \) using (22).

2. Let \( x^n_{s-1:t} = (x^n_{s-1}, \ldots, x^n_{t-K-1}, x^n_{t-K}, \ldots, x^n_t) \) and update the weight
   \[
   W^n_{s-1:t} \propto \hat{p}(y_t \mid x^n_{t-K-1:t-1}, y^n_{s-1:t-1}, \theta^n) \times W^n_{s-1:t-1}.
   \]
   where \( \hat{p} \) is defined in (26).

3. The particle simulation smoother. Generate \( k^n_j \sim \mathcal{M}(V^{-1}_{j, \theta^n}), j = t - 1, \ldots, t - K \), where \( V^n_{j, \theta^n} \) is given in (27), and set \( x^n_{s-1:t} = (x^n_{s-1}, \ldots, x^n_{t-K-1}, x^n_{t-K}, \ldots, x^n_t) \).

We use the prior \( f_{\theta} \) as the proposal, the above sampling becomes much more efficient than the simple rolling-window sampler as shown in Sect. 4.

3.2.2 Backward block sampling (step 2)

Before we describe the backward block sampling which generates a cloud of particles based on \( (x^n_{s+1:t}, \theta^n) \), we define the notation for the particle index as noted in the forward block sampling but in the reverse order. A ‘parent’ particle of \( x^n_j \) is chosen from \( x^n_{j+1} \) (not from \( x^n_{j+1} \)) and consequently \( a^n_{j+1} \) denotes its parent’s index. In this case, the relationship of \( a^n_{j+1} \) and \( k_j \) is given as follows:

\[
a^n_{j+1} = k_{j+1}, \quad j = s + K - 2, \ldots, s - 2.
\]

 Springer
For each $n$, we first generate $M$ particle paths, $x_{s-1:s+K-1}^n = (x_{s-1:s+K-1}^n, \ldots, x_{s+1:s+K-1}^n)$, and sample one path, $x_{s,t}^n$, from $x_{s-1:s+K-1}^n$ as noted below.

1. We generate $x_{s-1:s+K-1}^n \sim \pi(x_{s-1:s+K-1}^n \mid x_{s+1}^n, y_{s:t}, \theta^n)$.
   
   (a) Sample indices $k_j$ from $\{1, \ldots, M\}$ with probability $1/M$ ($j = s + K - 1, s + K - 2, \ldots, s - 1$) and set
   
   $$(n,k_1), \ldots, (n,k_{s+K-1}) = x_{s-1:s+K-1}^n, \quad (a_{s-1}^{k_2}, \ldots, a_{s+K-1}^{k_{s+K-2}}) = (k_1, \ldots, k_{s+K-1}),$$

   where $x_{s-1:s+K-1}^n$ is a current sample with the importance weight $W_{[s-1,t]}^n$.

   (b) Set $x_{s+K}^n = x_{s+K}^n$ for all $m$ according to the convention, and sample $x_{s+K-1}^n \sim q_{s+K-1,\theta^n}(\cdot \mid x_{s+K}^n, y_{s+K-1})$ for each $m \in \{1, \ldots, M\} \setminus \{k_{s+K-1}\}$. Let $j = s + K - 2$.

   (c) Sample $a_{j+1}^m \sim \mathcal{M}(V_{j+1,\theta^n})$ and $x_{j+1}^m \sim q_{j,\theta^n}(\cdot \mid x_{j+1}^m, y_j)$ for each $m \in \{1, \ldots, M\} \setminus \{k_j\}$ where $V_{j+1,\theta^n} = (V_{j+1,\theta^n}, \ldots, V_{j+1,\theta^n})$ and

   $$V_{j,\theta^n} = \frac{v_{j,\theta^n}(x_{j+1}^m, a_{j+1}^m)}{\sum_{i=1}^M v_{j,\theta^n}(x_{j+1}^i, a_{j+1}^i)},$$

   $$(n,m,j+1) = \left( n, a_{j+1}^m, x_{j+1}^m \right) = \frac{p(x_{j+1}^m \mid x_{j+1}^m, \theta)q_{j,\theta^n}(y_j \mid x_{j+1}^m, x_{j+1}^m)}{q_{j,\theta^n}(x_{j+1}^m \mid x_{j+1}^m, y_j)}, \quad m = 1, \ldots, M. \tag{29}$$

   (d) If $j > s - 1$, set $j \leftarrow j - 1$ and go to (c). Otherwise, sample $k_s^* \sim \mathcal{M}(V_{s,\theta^n})$ and obtain $k_j^* (j = s + 1, \ldots, s + K - 1)$ using (28).

3. Let $x_{s,t}^n = (x_{s}^{n,k_s^*}, \ldots, x_{s+K-1}^{n,k_s^*}, x_{s+K}^{n}, \ldots, x_{t}^{n})$ and compute its importance weight

   $$W_{[s,t]}^n \propto \begin{cases} 
   \frac{1}{\hat{p}(y_{s-1} \mid x_{s+K}^n, y_{s:t}, \theta^n)} W_{[s-1,t]}^n, & \text{if } \hat{p}(y_{s-1} \mid x_{s+K}^n, y_{s:t}, \theta^n) \neq 0, \\
   0, & \text{if } \hat{p}(y_{s-1} \mid x_{s+K}^n, y_{s:t}, \theta^n) = 0, \end{cases} \tag{31}$$

   where

   $$\hat{p}(y_{s-1} \mid x_{s+K}^n, y_{s:t}, \theta^n) = \frac{1}{M} \sum_{m=1}^M v_{s-1,\theta^n}(x_{s-1}^m, x_{s}^{n,k_s^*}), \tag{32}$$

   and $\hat{p}(y_{s-1} \mid x_{s+K}^n, y_{s:t}, \theta^n)\text{-}1$ can be seen as the estimate of the intractable incremental weight $p(y_{s-1} \mid x_{s+K}^n, y_{s:t}, \theta^n)\text{-}1$ in (18) for the idealized double-block sampler.

---

5 Note that we need to generate $x_{s-1}^n$ to compute $\hat{p}$ in (32).
3. Implement the particle simulation smoother to sample \( (k_s^*, k_{s+1}^*, \ldots, k_{s+K-1}^*) \) jointly. Generate \( k_j^* \sim \mathcal{M}(\tilde{V}_{j,0^n}^1:1^M) \), \( j = s + 1, \ldots, s + K - 1 \), recursively where

\[
\tilde{V}_{j,0^n}^m = \frac{V_{j,0^n}^m p(x_{j-1}^{k_j^*-1} | x_j^m, \theta^n)}{\sum_{t=1}^M V_{j,0^n}^t p(x_{j-1}^{k_j^*-1} | x_j^t, \theta^n)}, \quad m = 1, \ldots, M. \tag{33}
\]

and set \( x^n_{s:t} = (x^n_s, \ldots, x^n_{s+K-1}, x^n_{s+K}, \ldots, x^n_t) \).

---

**Algorithm 3 (Step 2) : Practical double-block sampler**

Let \( (x^n_{s-1:t}, \theta^n) \) denote the sample from \( \pi(x_{s-1:t}, \theta | y_{s-1:t}) \) with the weight \( W^n_{[s-1:t]} \) \( (n = 1, \ldots, N) \).

1. We generate \( x^n_{s-1:s+K-1} \sim \pi(x^n_{s-1:s+K-1} \mid x^n_{s+K:t}, y_{s:t}, \theta^n) \) \( (s \leq t = K - 1) \) as follows:

   (a) Sample \( k_j \) from \( \{1, \ldots, M\} \) with probability \( 1/M \) \( (j = s + K - 1, \ldots, s - 1) \) and set

   \[
   (x^n_{s-1}, \ldots, x^n_{s+K-1}) = x^n_{s-1:s+K-1}, \quad (a^n_{s-1}, \ldots, a^n_{s+K-1}) = (k^n_{s-1}, \ldots, k^n_{s+K-1}).
   \]

   (b) Set \( x^n_{s+K} = x^n_{s+k} \) and sample \( x^n_{s+K} \sim q_{s+k-1, \theta^n} (\cdot \mid x^n_{s+K}, y_{s+K-1}) \), \( m \in \{1, \ldots, M\} \setminus \{k^n_{s+K}\} \), \( j = s + K - 2 \).

   (c) Sample \( a^n_{j+1} \sim \mathcal{M}(\tilde{V}_{j,\theta^n}^1:1^M) \) and \( x^{n,a^n_{j+1}}_j \sim q_{j, \theta^n} (\cdot \mid x^{n,a^n_{j+1}}_{j+1}, y_j) \), \( m \in \{1, \ldots, M\} \setminus \{k^n_j\} \) where \( V_{j+1, \theta^n}^1 \) is given by (29).

   (d) If \( j > s - 1 \), set \( j \leftarrow j - 1 \) and go to (c). Otherwise, sample \( k^n_j \sim \mathcal{M}(\tilde{V}_{j,\theta^n}^1:1^M) \) and obtain \( k^n_j \) \( (j = s + 1, \ldots, s + K - 1) \) using (28).

2. Let \( x^n_{s:t} = (x^n_s, \ldots, x^n_{s+K-1}, x^n_{s+K}, \ldots, x^n_t) \) and update the weight

\[
W^n_{[s:t]} \propto \hat{p}(y_{s-1} | x^n_{s+K}, y_{s:t}, \theta^n)^{-1} \times W^n_{[s-1:t]}, \tag{34}
\]

where \( \hat{p} \) is defined in (32). If \( \hat{p} = 0 \), set \( W^n_{[s:t]} = 0 \).

3. The particle simulation smoother. Generate \( k_j^* \sim \mathcal{M}(\tilde{V}_{j,\theta^n}^1:1^M) \), \( j = s + 1, \ldots, s + K - 1 \), where \( \tilde{V}_{j,\theta^n}^m \) is given in (33), and set \( x^n_{s:t} = (x^n_s, \ldots, x^n_{s+K-1}, x^n_{s+K}, \ldots, x^n_t) \).

Figure 2 illustrates an example with \( K = 2, M = 4 \) and the current sample \( (x^n_{s-1:t}, \theta^n) \).

1. (a) Sample indices \( k_{s+1}, k_s, k_{s-1} \) from \{1, 2, 3, 4\} with probability \( 1/4 \) and suppose \( k_{s+1} = 1, k_s = 1, k_{s-1} = 1 \). We set \( (x^n_{s-1}, x^n_s, x^n_{s+1}) = x^n_{s-1:s+1} \) (with the rectangle) and \( (a^n_{s-1}, a^n_s, a^n_{s+1}) = (1, 1, 1) \).
(b) Set $x_{s+2}^n=x_{s+2}^n$ for all $m$ (with the thick black rectangle), and sample $x_{s+1}^n \sim q_{s+1,0^n}(\cdot \mid x_{s+1}^n, y_{s+1})$ for $m \in \{2, 3, 4\}$ (with the black circle).

(c) Sample $a_{s+1}^m \sim \mathcal{M}(V_{s+1}^{1:4}, \theta^n)$ and suppose $a_{s+1}^2 = 1, a_{s+1}^3 = 3, a_{s+1}^4 = 3$.

Generate $x_{s+1}^n \sim q_s, \theta^n(\cdot \mid x_{s+1}^n, y_s)$ for $m \in \{2, 3, 4\}$.

(d) Sample $a_s^m \sim \mathcal{M}(V_{s}^{1:4}, \theta^n)$ and suppose $a_s^2 = 2, a_s^3 = 2, a_s^4 = 4$. Generate $x_{s-1}^n \sim q_{s-1,0^n}(\cdot \mid x_{s}^n, y_{s-1})$ for $m \in \{2, 3, 4\}$.

(e) Sample $K_s^* \sim \mathcal{M}(V_{s}^{1:4}, \theta^n)$ and suppose $K_s^* = 2$. Using (28), we obtain $k_{s+1} = 1$, and select $(x_{s}^{2}, x_{s+1}^{n+1})$ with red lines.

2. Let $x_{s:t}^n=(x_{s}^n, x_{s+1}^n, x_{s+2}^n, \ldots, x_{t}^n)$ and compute its importance weight.

3. Implement the particle simulation smoother to sample $(k_s^*, k_{s+1}^*)$ jointly. Generate $k_{s+1}^* \sim \mathcal{M}(V_{s+1}^{1:4}, \theta^n)$, and suppose $k_{s+1}^* = 2$. We set $x_{s:t}^n=(x_{s}^{n+2}, x_{s+1}^n, x_{s+2}^n, \ldots, x_{t}^n)$.

**Remark 2** In Algorithm 3, we assume we can evaluate the density $p(x_{j-1} \mid x_j, \theta)$ defined as

$$p(x_{s-1} \mid x_s, \theta) = \frac{\mu_\theta(x_{s-1}) f_\theta(x_s \mid x_{s-1})}{\mu_\theta(x_s)}.$$

**Remark 3** In the simple rolling-window sampler, we reweighted the particles according to the likelihood $q_\theta(y_{s-1} \mid x_{s-1}, x_s)$ in Step 2, while the unbiased estimate of the conditional likelihood $\hat{p}(y_{s-1} \mid x_{s+K}, y_{s:t}, \theta^n)$ is used in the practical double-block sampler. Algorithm 3 substantially improves the weight degeneracy since we condition on $y_{s:t}$ and integrate out $(x_{s-1}, \ldots, x_{s+K-1})$. 
Remark 4 As we apply the nested SMC algorithm in which one generates $M$ random values for each particle, the computational cost is larger than Algorithms 1 and 2. The cost for the three discussed algorithms are $O(N \times T)$, $O(K \times N \times T)$, and $O(K \times M \times N \times T)$, respectively. Using Algorithm 1, however, leads to serious weight degeneracy as confirmed in Sect. 4.1, and running Algorithm 2 is not realistic since the proposal distribution tends to be not available, so Algorithm 3 is the most realistic for the rolling-window analysis. In Sect. 4 we see that $M$ needs to be 100 or 300 for satisfactory performance.

3.3 Sequential MCMC estimation without rolling the window

In the above discussion, it is implicitly assumed that the initial particles approximating $\pi(x_{1:L+1}, \theta \mid y_{1:L+1})$ are obtained. To sample from this initial posterior distribution, using MCMC-based methods is straightforward as in the warm-up period for the practical filtering described in Polson et al. (2008). Moreover, we could simply use MCMC samples from the initial posterior distribution. However, based on our proposed method for the rolling estimation, we can obtain samples of $x_{1:L+1}$ and $\theta$ sequentially, simply by skipping Step 2. The advantage of using our SMC-based method is that we can obtain the estimate of marginal likelihood $p(y_{1:L+1})$ as a by-product (the initializing algorithm and the marginal likelihood estimator are described in detail in the Supplementary Material B.). This initializing algorithm can be used for the ordinary sequential learning of $\pi(x_{1:t}, \theta \mid y_{1:t})$ ($t = 1, \ldots, T$). We note that this approach is derived from the particle Gibbs scheme in Andrieu et al. (2010), and hence our approach is different from that of SMC^2 which applies the particle MH scheme as noted in Chopin et al. (2013) and Fulop and Li (2013).

4 Illustrative examples

This section demonstrates the efficiencies of our proposed algorithm using two illustrative examples. The simple rolling-window sampler suffers from the serious weight degeneracy problem, while (the idealized and the practical) double-block samplers overcome such difficulties. To evaluate the weight degeneracy in each of Steps 1 and 2, we define two ratios:

$$R_{1t} = \frac{\text{ESS}_{[s-1:t]}^{x_{s-1:t}}}{\text{ESS}_{[s-1:t-1]}^{x_{s-1:t-1}}}, \quad R_{2t} = \frac{\text{ESS}_{[s:t]}^{x_{s:t}}}{\text{ESS}_{[s-1:t]}^{x_{s-1:t}}}.$$ (35)

The ratio $R_{1t}$ measures the relative change of ESS in Step 1 after adding $y_t$ when compared with that of the previous step. If the distribution of particles is close to the posterior distribution from which we aim to sample in the step, $R_{1t}$ would be close to 1. On the other hand, in the presence of the weight degeneracy problem, it will be close to 0. Similarly, the ratio $R_{2t}$ measures the relative change of ESS in Step 2 after removing $y_{s-1}$ compared with that of the previous step.
Table 1  The number of resampling steps of three samplers

|        | Simple | $K$ | Idealized | Practical |
|--------|--------|-----|-----------|-----------|
| $M: 100$ |       |     |           | 100       | 300       | 500       |
| 1027   | 1      | 48  | 104       | 71        | 61        |
| 2      | 8      | 74  | 33        | 23        |
| 3      | 7      | 74  | 31        | 22        |
| 5      | 6      | 72  | 32        | 22        |
| 10     | 5      | 69  | 31        | 23        |

4.1 Linear Gaussian state space model

We first consider the following univariate linear Gaussian state space model:

$$y_t = x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2), \quad t = 1, \ldots, 2000$$

$$x_{t+1} = \mu + 0.25(x_t - \mu) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, 2\sigma^2), \quad t = 1, \ldots, 2000,$$

$$x_1 = \mu + \frac{\eta_0}{\sqrt{1 - 0.25^2}}, \quad \eta_0 \sim \mathcal{N}(0, 2\sigma^2),$$

where $\theta = (\mu, \sigma^2)'$ is a parameter vector. We adopt weak conjugate priors, $\mu \mid \sigma^2 \sim \mathcal{N}(0, 10\sigma^2)$ and $\sigma^2 \sim \mathcal{IG}(5/2, 0.05/2)$ where $\mathcal{IG}(a, b)$ denotes an inverse gamma distribution with shape parameter $a$ and scale parameter $b$. The rolling estimation is conducted with a window $[t - 999, t]$, $t = 1001, \ldots, 2000$ and $N = 1000$ using the particle rolling MCMC with and without the double-block sampling. We choose $K = 1, 2, 3, 5$ and 10 to investigate the effect of the block size. Since an idealized double-block sampler is feasible in the linear Gaussian state space model, we compare the following three samplers:

1. Simple rolling-window sampler (as a benchmark).
2. Idealized double-block sampler.
3. Practical double-block sampler with $M = 100, 300$ and 500.

Table 1 shows the number of resampling steps for three samplers. For the simple rolling-window sampler, the resampling steps are triggered 1027 times, while they are drastically reduced for the double-block samplers. They decrease as we increase $K$ where the magnitude of the reduction is largest at $K = 2$. For $K = 2$, they are around 0.8% and 7.2% of the simple rolling-window sampler for the idealized and practical double-block samplers respectively. In addition, the number of resampling steps of the practical double-block sampler decreases to that of the idealized double-block sampler as $M$ increases.

Figure 3 shows histograms of $R_{1t}$ and $R_{2t}$ for the simple rolling-window sampler and the practical double-block sampler with $K = 2$ and $M = 100$. The ratios $R_{1t}$ and $R_{2t}$ measure the relative magnitude of the effective sample size in Step 1 and Step 2 after adding $y_t$ and removing $y_{t-1}$ respectively when compared with that of the previous step at time $t$. The $R_{1t}$ values for the practical double-block sampler are larger and less dispersed compared with those for the simple rolling-window sampler, suggesting that the forward block sampling is more efficient. In addition, the $R_{2t}$ values
for the practical double-block sampler are much larger and much less dispersed than those for the simple rolling-window sampler, which implies that the backward block sampling is much more efficient.

Further, the scatter plots of \( R_{1t} \) and \( R_{2t} \) are shown in Fig. 4 for two sampling methods. These results demonstrate that our practical double-block sampler is more efficient at both Steps 1 and 2 of each rolling step.

Table 2 shows the summary statistics of \( R_{1t} \) and \( R_{2t} \). The average of \( R_{1t} \) for the practical double-block sampler is slightly larger than that for the simple rolling-window sampler, but the standard deviation for the former is less than half of that for the latter. Moreover, the average of \( R_{2t} \) for the double-block sampling is six times larger than that for the simple sampling, while the standard deviation for the former is approximately half of that for the latter. Thus, the practical double-block sampler drastically alleviate the weight degeneracy compared with the simple rolling-window sampler.

Finally, to assess the accuracy of the practical double-block sampler (with \( K = 2 \) and \( M = 100 \)), we compare the estimation results with their corresponding analytical solutions. The particles are ‘refreshed’ in the MCMC update step so that the approximation errors do not accumulate over time. In Fig. 5, the algorithm seems to correctly capture both means and 95% credible intervals of the target posterior distribution. In Fig. 6, true log marginal likelihoods and their estimates are shown in with errors. The estimation errors are very small overall, implying that the proposed algorithm estimates the marginal likelihood \( p(y_{t-999:t}) \) accurately for \( t = 1001, \ldots, 2000 \).
Fig. 4 The scatter plot of $R_{2t}$ versus $R_{1t}$ ($t = 1001, \ldots, 2000$) for the simple rolling-window sampler (blue plus) and the practical double-block sampler with $K = 2$ and $M = 100$ (red circle) (color figure online)

Table 2 Summary statistics of $R_{1t}$ and $R_{2t}$ for the simple rolling-window sampler and the practical double-block sampler ($K = 2$, $M = 100$)

|          | Method   | Mean  | Std. dev. |
|----------|----------|-------|-----------|
| $R_{1t}$ | Simple   | 0.862 | 0.145     |
|          | Practical| 0.975 | 0.057     |
| $R_{2t}$ | Simple   | 0.161 | 0.139     |
|          | Practical| 0.970 | 0.068     |

4.2 Realized stochastic volatility model

This subsection considers the RSV model given by (3)–(7) where the idealized double-block sampler is not feasible. For the static parameter $\theta = (\mu, \phi, \sigma_\eta^2, \xi, \sigma_u^2, \rho)'$, we assume the prior distributions as in Takahashi et al. (2009):

$$\frac{\phi + 1}{2} \sim B(20, 1.5), \ c \sim N(0, 10), \ \sigma_u^2 \sim IG(5/2, 0.05/2),$$

$$\Sigma \sim IW(5, \Sigma_0), \ \Sigma_0 = \left( 5 \begin{bmatrix} 1 & -0.3 \times 0.1 \\ -0.3 \times 0.1 & 0.01 \end{bmatrix} \right)^{-1}.$$

using the transformation

$$\sigma_e = \exp(\mu/2), \ c = \xi + \mu, \ \Sigma = \begin{bmatrix} \sigma_e^2 & \rho \sigma_e \sigma_\eta \\ \rho \sigma_e \sigma_\eta & \sigma_\eta^2 \end{bmatrix}.$$
Fig. 5  True posterior means and 95% credible intervals (dotted black) with their estimates (solid red) for $\mu$ and $\sigma^2$ using the practical double-block sampler (color figure online).

Fig. 6  Top: true log marginal likelihoods $\log p(y_t|\gamma_{t-999:t})$ (dotted black) and their estimates (solid red) by the practical double-block sampler. Bottom: estimation errors $\log \hat{p}(y_t|\gamma_{t-999:t}) - \log p(y_t|\gamma_{t-999:t})$ for $t = 1001, \ldots, 2000$ (color figure online).
where $B(a, b)$ and $IW(r, S)$ denote a beta distribution with parameters $(a, b)$, and an inverse Wishart distribution with $r$ degrees-of-freedom and the scale matrix $S$ respectively. For $y_1$ and $y_2$, we use Standard and Poor’s (S&P) 500 index data, which are obtained from the Oxford-Man Institute Realized Library\(^6\) created by Heber et al. (2009) (see Shephard and Sheppard (2010) for details). The initial estimation period is from January 1, 2000 ($t = 1$) to December 31, 2007 ($t = 1988$) with $L + 1 = 1988$. The rolling estimation started after this initial sample period and moved the window until December 30, 2008 ($T = 2248$). Thus, the first estimation period is before the financial crisis caused by the bankruptcy of Lehman Brothers and the last estimation period includes the crisis.

We first implement the simple rolling-window sampler. If the ESS is less than the threshold ($0.5 \times N$), the particles are refreshed with the MCMC update 10 times. (see Takahashi et al. (2009) for the details of the MCMC sampling). We set $N = 1000$ and construct the proposal density $q_{t, \theta}(x_t \mid x_{t-1}, y_{s-1:t})$ based on the normal mixture approximation (see Omori et al. (2007)), which is expected to improve the weight degeneracy. Table 3 presents a summary of $R_{1t}$ and $R_{2t}$. As expected, $R_{2t}$'s are low, so the update with MCMC kernel should be implemented in almost every step. The results for $R_{1t}$'s also indicate that the ESS will be often less than the threshold to resample all the particles. In fact, due to these problems, the resampling steps are implemented 271 times for 260 data windows.

Next, we implement the practical double-block sampler with\(^7\) $M = 300$ and $N = 1000$. Further we always implement 10 MCMC iterations below unless otherwise stated. As a proposal density, we simply use a prior density $q_{t, \theta}(x_t \mid x_{t-1}, y_{s-1:t}) = f_{\theta}(x_t \mid x_{t-1})$ to demonstrate that the practical double-block sampler improves even when using the simple proposal. The summary statistics of $R_{1t}$ and $R_{2t}$ are shown in Table 4 where we use $K = 5, 10$ and $15$. In contrast to the simple rolling-window algorithm, both means are close to 1 demonstrating that our proposed algorithm succeeded in overcoming the weight degeneracy problem. As $K$ increases, $R_{1t}$ and $R_{2t}$ become larger and less dispersed, but the difference becomes smaller for $K = 10$ and $K = 15$.

Figure 7 shows the trace plot of estimated posterior means and 95% credible intervals for $\theta = (\mu, \phi, \sigma^2_\eta, \xi, \sigma^2_u, \rho)'$ from December 31, 2007 ($t = 1988$) to December 30, 2016 ($t = 4248$). From the rolling estimation results, we are able to observe the transition of the economic structure and the effect of the financial crisis ($t = 2150, \ldots, 2213$ correspond to September, October and November in 2008). The posterior distribution of $\mu$ seems to be stable before $t = 4000$ (January 7, 2016), but its mean and 95% intervals decrease after $t = 4000$. The average level of log volatility

---

\(^6\) The data are downloaded at [http://realized.oxford-man.ox.ac.uk/data/download](http://realized.oxford-man.ox.ac.uk/data/download).

\(^7\) We also tried using other values of $M$ but the computation time is the shortest with $M = 300$. 

---

**Table 3** Summary statistics for $R_{1t}$ and $R_{2t}$ ($t = 1988, \ldots, 2248$) for the simple rolling-window sampler

|       | Mean   | Median | Std. dev. |
|-------|--------|--------|-----------|
| $R_{1t}$ | 0.837  | 0.912  | 0.193     |
| $R_{2t}$ | 0.227  | 0.197  | 0.176     |
Table 4  Summary statistics for $R_{1t}$ and $R_{2t}$

\[ (t = 1988, \ldots, 4248) \] using the practical double-block sampler

| $K$ | Mean | Median | Std. dev. |
|-----|------|--------|-----------|
| 5   | 0.981 | 0.995  | 0.058     |
| 10  | 0.985 | 0.996  | 0.053     |
| 15  | 0.986 | 0.997  | 0.055     |
| 5   | 0.983 | 0.993  | 0.044     |
| 10  | 0.988 | 0.994  | 0.036     |
| 15  | 0.988 | 0.994  | 0.035     |

Fig. 7  Trace plot of estimated posterior means and 95% credible intervals for parameters for S&P500 return in RSV model (from December 31, 2007 to December 30, 2016) using the practical double-block sampler with 5 (dotted blue) and 10 (solid red) iterations for the MCMC step (color figure online)

Figure 8 shows three cumulative computation times (wall time) for the same period corresponding to $K = 5, 10$ and 15. The computation times with $K = 5$ and $K = 15$ are longer than that with $K = 10$. This finding implies that, when $K = 5$, the effect of the blocking is not sufficient to reduce the path dependence between $x_t$ and $x_{t-K-1}$ (similarly, $x_{s-1}$ and $x_{s+K}$). When $K = 15$, the Monte Carlo error in the local conditional SMC increased the variance of the importance weights even though there started to decrease sharply toward the end of the sample period. The autoregressive parameter, $\phi$, continues to decrease throughout the sample period indicating that the latent log volatility becomes less persistent. The variances, $\sigma^2_\eta$ and $\sigma^2_u$, of error terms in the state equation and the measurement equation of the log realized volatility continue to increase, while the bias adjustment term, $\xi$, and the leverage effect, $\rho$, become closer to zero during the sample period. The leverage effects in the stock market are weaker after the financial crisis.
is a certain decrease in the variance due to the increase in $K$ (we shall see more details in Sect. 5).8

Finally, in Fig. 9, we investigate the effect of the number of iterations in the MCMC steps on the estimation accuracy of the posterior distribution function of $\theta$ for the proposed sampling algorithm. The estimation period is from January 1, 2000 to December 31, 2007 ($t = 1, \ldots, 1988$). First, the MCMC sampling is conducted to obtain the accurate estimates of the distribution functions (solid gray). Then, we apply our practical double-block sampler with $K = 10$, $M = 300$ and $N = 1000$ for three cases: one, five and ten MCMC updates. Among three cases, the estimates obtained by 5 or 10 iterations are close to those obtained by the exact MCMC sampling. If only one iteration is performed in the MCMC update step, the estimation results are found to be inaccurate because the MCMC iterations not only diversify the particles but also correct approximation errors introduced by the particle algorithm, which basically update only a part of the vector $x_{t-1}^n$. The estimation errors for the distribution function of $\mu$ are most serious, probably because the mixing property of MCMC sampling in the RSV model is poor especially with respect to $\mu$ as discussed in the numerical studies of Takahashi et al. (2009). Thus, these results suggest that MCMC iterations should be implemented a sufficient number of times in the MCMC update steps such that the particles can trace the correct posterior distributions.

---

8 Also see Supplementary Material C for the comparison of the computation time of the practical double-block sampler with those of the MCMC and the particle MCMC.
5 Theoretical justification

In this section, we provide theoretical justifications of our proposed algorithm in Sect. 3. We prove that our posterior density is obtained as a marginal density of the artificial target density.

5.1 Forward block sampling

The artificial target density and its marginal density. We prove that our posterior density of $(x_{s-1:t}^n, \theta^n)$ given $y_{s-1:t}$ is obtained as a marginal density of the artificial target density in the forward block sampling. The superscript $n$ will be suppressed for simplicity below.

In Step 1-1(a) of Sect. 3.2.1, the probability density function of $(x_{t-K}^{k_{t-K}}, \ldots, x_{t-1}^{k_{t-1}}) = x_{t-K:t-1}$ and $(a_{t-K}^{k_{t-K}+1}, \ldots, a_{t-1}^{k_{t-1}})$ given $(x_{t-K-1}, \theta)$ and $y_{t-K:t-1}$ is

$$p(x_{t-K:t-1}, a_{t-K}^{k_{t-K}+1}, \ldots, a_{t-1}^{k_{t-1}} \mid x_{t-K-1}, y_{t-K:t-1}, \theta) = \frac{\pi(x_{t-K:t-1} \mid x_{t-K-1}, y_{t-K:t-1}, \theta)}{M^K}.$$ (39)

Let $a_j^{1:M} = (a_j^1, \ldots, a_j^M)$, $x_j^{1:M} = (x_j^1, \ldots, x_j^M)$ and $a_j^{-k_{j+1}} = a_j^{1:M} \setminus a_j^{k_{j+1}} = a_j^{1:M} \setminus k_j$ for $j = t - K, \ldots, t - 1$ where we note $a_j^{k_{j+1}} = k_j$ and $k_t = 1$ in
(21). Further, let \( a_{t-K,t-1}^1 = \{a_{t-K}^1, \ldots, a_{t-1}^1\} \), and \( x_j^{-k_j} = \{x_j^1, \ldots, x_j^{a_j^M}\} \setminus x_j^{k_j} \). Then, in 1(b), 1(c) and 1(d) of Step 1, given \( x_{t-K-1}, (x_{t-K}^1, \ldots, x_{t-1}^1) = x_{t-K,t-1} \) and \((a_{t-K}^1, \ldots, a_{t-1}^1) = (k_{t-K}, \ldots, k_{t-1})\), the probability density function of all variables is defined as

\[
\psi_\theta \left( x_{t-K}^1, \ldots, x_{t-1}^1, a_{t-K}^1, \ldots, a_{t-1}^1, k_{t-K}^*, \ldots, k_{t-1}^* \mid x_{t-K-1}, y_{t-K-1} \right) = \prod_{m=1}^{M} q_{t-K, m}^{x_{t-K}^m} \times \prod_{j=t-K+1}^{t-1} \prod_{m=1}^{M} V_{t-1, \theta}^{a_j^m} q_{j, \theta} (x_j^m \mid x_{j-1}^{a_j^{m-1}}, y_j) 
\times q_{t, \theta} (x_t^1 \mid x_{t-1}^1, y_t) \times V_{t-1, \theta}^{k_t^*}
\]

(40)

In Step 1-2, we multiply \( W_{s-1:t-1} \) by \( \hat{p}(y_t \mid x_{t-K}^n, y_{s-1:t-1}, \theta^n) \) to adjust the importance weight for \( W_{s-1:t} \). Let \( x_{t-K,t-1}^1 = (x_{t-K}^1, \ldots, x_t^1) \) and \( a_{t-K,t-1}^1 = (a_{t-K}^1, \ldots, a_{t-1}^1) \). Our artificial target density (before the particle smoother step) is written as

\[
\hat{\pi} (x_{s-1:t-1} \mid x_{t-K-1}^1, x_{t-K,t-1}^1, a_{t-K,t-1}^1, k_{t-K}^*, \ldots, k_{t-1}^* \mid y_{s-1:t}) = \pi (x_{s-1:t-1} \mid k_{t-K}^{1:t}, \theta \mid y_{s-1:t}) 
\times \psi_\theta (x_{t-K}^1, \ldots, x_{t-1}^1, a_{t-K}^1, \ldots, a_{t-1}^1, k_{t-K}^*, \ldots, k_{t-1}^* \mid x_{t-K-1}, y_{t-K-1}) 
\times \hat{p}(y_t \mid x_{t-K-1}, y_{t-1:t-1}, \theta) 
\times \pi (x_{t-1:t-1} \mid \theta \mid y_{t-1:t-1}) 
\times \left( \prod_{m=1}^{M} q_{t-K, m}^{x_{t-K}^m} \times \prod_{j=t-K+1}^{t-1} \prod_{m=1}^{M} V_{t-1, \theta}^{a_j^m} q_{j, \theta} (x_j^m \mid x_{j-1}^{a_j^{m-1}}, y_j) 
\times q_{t, \theta} (x_t^1 \mid x_{t-1}^1, y_t) \times V_{t-1, \theta}^{k_t^*} \right) 
\times \frac{\hat{p}(y_t \mid x_{t-K-1}, y_{t-1:t-1}, \theta)}{p(y_t \mid y_{s-1:t-1})}.
\]

(41)

Note that \( p(y_t \mid y_{s-1:t-1}) \) is the normalizing constant of this target density, which will be shown in Proposition 5.2. The proposed forward block sampling is justified by proving that the marginal density of \((x_{s-1}, \ldots, x_{t-K-1}, x_{t-K}^{k_{t-K}^*}, \ldots, x_t^{k_t^*}, \theta)\) in the above artificial target density \(\hat{\pi}\) is \( \pi (x_{s-1}, \ldots, x_{t-K-1}, x_{t-K}^{k_{t-K}^*}, \ldots, x_t^{k_t^*}, \theta \mid y_{s-1:t}) \).

**Proposition 5.1** The artificial target density \(\hat{\pi}\) for the forward block sampling can be written as

\(\hat{\pi}\) Springer
\( \hat{\pi}(x_{s-1:t-K-1}, x_{t-K}^{1:M}, a_t^{1:M}, k_t^*, \theta \mid y_{s-1:t}) \)
\[ = \frac{\pi(x_{s-1:t-K-1}, x_t^{1-K}, \ldots, x_t, \theta \mid y_{s-1:t})}{M^{K+1}} \times \prod_{m=1}^{M} q_{t-K, \theta}(x_t^m \mid x_{t-K-1}, y_{t-K}) \]
\times \prod_{j=t-K+1}^t \prod_{m=1 \atop m \neq k_t^*}^M \int_{j-1, \theta}^j \int_{j}^j \int_{j}^j \pi(x_{s-1:t}) \mid x_{s-1:t-K}, x_t^{1-K}, \ldots, x_t^*, \theta \mid y_{s-1:t}) \]

and the marginal density of \((x_{s-1:t-K-1}, x_{t-K}^{1:M}, a_t^{1:M}, k_t^*, \theta) \) is \( \pi(x_{s-1:t-K-1}, x_t^{1-K}, \ldots, x_t, \theta \mid y_{s-1:t}) \).

**Proof** See Supplementary Material A.

Proposition 5.1 implies that we can obtain a posterior random sample \((x_{s-1:t}, \theta) \) given \( y_{s-1:t} \) (with the importance weight \( W_{[s-1:t]} \)) by sampling from the artificial target distribution \( \hat{\pi} \). This justifies our proposed forward block sampling scheme.

**Properties of the incremental weight.** We consider the mean and variance of the (unnormalized) incremental weight, \( \hat{\rho}(y_t \mid y_{s-1:t-1}, x_{t-K-1}, \theta) \). Proposition 5.2 shows that this weight can be considered an unbiased estimator.

**Proposition 5.2** If
\[ (x_{s-1:t-K-1}, x_t^{1-K}, \ldots, x_t^{k_t-1}, \theta) \sim \frac{\pi(x_{s-1:t-K-1}, x_t^{k_t-1}, \ldots, x_t^{k_t-1}, \theta \mid y_{s-1:t-1})}{M^K} \]
and
\[ (x_t^{-k_t-1}, \ldots, x_t^{-k_t-1}, x_t^{1:M}, a_t^{1-K}, \ldots, a_t^{1-M}, k_t^*) \sim \psi_\theta \]

where \( \psi_\theta \) is given in (40), then
\[ E[\hat{\rho}(y_t \mid x_{t-K-1}, y_{s-1:t-1}) \mid y_{s-1:t}] = E[p(y_t \mid x_{t-K-1}, y_{s-1:t-1}) \mid x_{t-K-1}, y_{s-1:t}, \theta] \]
\[ = p(y_t \mid y_{s-1:t-1}). \]

**Proof** See Supplementary Material A.

This shows that the incremental weight \( \hat{\rho}(y_t \mid x_{t-K-1}, y_{s-1:t-1}, \theta) \) is an unbiased estimator of the conditional likelihood \( p(y_t \mid x_{t-K-1}, y_{s-1:t-1}, \theta) \) given \( (x_{t-K-1}, \theta) \). It is also an unbiased estimator of the marginal likelihood \( p(y_t \mid y_{s-1:t-1}) \) unconditionally, which implies that \( p(y_t \mid y_{s-1:t-1}) \) is a normalizing constant for the artificial target density \( \hat{\pi} \).

Further, from the law of total variance, we obtain the decomposition of the variance as follows.
\[
\text{Var}[\hat{p}(y_t | x_{t-K-1}, y_{s-1:t-1}, \theta) | y_{s-1:t}] = \text{Var}[p(y_t | x_{t-K-1}, y_{s-1:t-1}, \theta) | y_{s-1:t}] + E \left[ \text{Var}[\hat{p}(y_t | x_{t-K-1}, y_{s-1:t-1}, \theta) | x_{s-1:t-K-1}, y_{s-1:t}, \theta] \right].
\]

The variance of the incremental weight consists of two components, including variance of the conditional likelihood and (expected) variance which is introduced using \(M\) particles to approximate the conditional likelihood. This decomposition identifies factors that influence the ESS of the particles. Regarding the first component, for any positive integers, \(K_1, K_2\), with \(K_1 < K_2\), the following inequality holds:

\[
\text{Var}[p(y_t | x_{t-K_1-1}, y_{s-1:t-1}, \theta)] \geq \text{Var}[p(y_t | x_{t-K_2-1}, y_{s-1:t-1}, \theta)],
\]

which is a straightforward result from the law of total variance for \(p(y_t | x_{t-K_1-1}, y_{s-1:t-1}, \theta)\) using

\[
E \left[ p(y_t | x_{t-K_1-1}, y_{s-1:t-1}, \theta) | x_{s-1:t-K_2-1}, \theta \right] = p(y_t | x_{t-K_2-1}, y_{s-1:t-1}, \theta). \quad (43)
\]

On the other hand, the second component may become large as \(K\) increases, but it is expected to be controlled by changing the number of particles \(M\).

### 5.2 Backward block sampling

The artificial target density and its marginal density. This subsection proves that our posterior density of \((x^n_{s:t}, \theta^n)\) given \(y_{s:t}\) is obtained as a marginal density of the artificial target density in the backward block sampling. The superscript \(n\) will be suppressed for simplicity below.

In Step 2-1(a) of Sect. 3.2.2, the probability density function of \((x^{k_{s-1}}, \ldots, x^{k_{s+K-1}}_{s-1}) = x_{s-1:s+K-1}^{k_{s-1}}\) and \((a^{k_{s-1}}, \ldots, a^{k_{s+K-2}}_{s+K-1})\) given \((x_{s+K}, \theta)\) and \(y_{s-1:t}\) is

\[
p(x_{s-1:s+K-1}^{k_{s-1}}, a_{s-1}^{k_{s-1}}, \ldots, a_{s+K-1}^{k_{s+K-2}} | x_{s+K}, y_{s-1:t}, \theta) = \frac{\pi(x_{s-1:s+K-1} | x_{s+K}, y_{s-1:t}, \theta)}{M^{K+1}}. \quad (44)
\]

In 1(b), 1(c) and 1(d) of Steps 2, given \(x_{s+K}, (x^{k_{s-1}}, \ldots, x^{k_{s+K-1}}_{s-1}) = x_{s-1:s+K-1}^{k_{s-1}}\), \((a^{k_{s-1}}, \ldots, a^{k_{s+K-2}}_{s+K-1}) = (k_{s-1}, \ldots, k_{s+K-1})\) and \(y_{s-1:s+K-1}\), the probability density function of all variables is defined as

\[
\psi_\theta(x^{k_{s-1}}, \ldots, x^{k_{s+K-2}}_{s+K-1}, a^{k_{s-1}}, \ldots, a^{k_{s+K-2}}_{s+K-1} | x_{s-1:s+K-1}^{k_{s-1}}, a_{s-1}^{k_{s-1}}, \ldots, a_{s+K-1}^{k_{s+K-2}}, y_{s-1:s+K-1}) = \prod_{m=1}^{\tilde{M}} q_{s+K-1, \theta}(x^{m}_{s+K-1} | x_{s+K}, y_{s+K-1}) \times \prod_{j=s-1}^{s+K-2} \prod_{m=1}^{M} \prod_{j+1,0} \tilde{M}^j q_{j, \theta}(x^{m}_{j+1}, y_{j}). \quad (45)
\]

In Step 2-2, we divide \(W_{[s-1:t]}\) by \(\hat{p}(y_{s-1} | x^n_{s+K}, y_{s:t}, \theta^n)\) to adjust the importance weight for \(W_{[s,t]}\). Similarly to the discussion in Sect. 5.1, we consider an extended space with the artificial target density written as
where \( p(y_{s-1} \mid y_{s:t})^{-1} \) is the normalizing constant of this target density as shown in Proposition 5.4. Below we state Proposition 5.3 for the backward block sampling, which correspond to Proposition 5.1 for the forward block sampling.

**Proposition 5.3** The artificial target density \( \tilde{\pi} \) for the backward block sampling can be rewritten as

\[
\tilde{\pi}(x_{s-1:M}^1, x_{s:M:t}, a_{s:M}^1, k_s, k_s^*, \theta \mid y_{s-1:t}) = \pi(x_{s-1:M}^1, \ldots, x_{s:M}^1, x_{s:M:t}, \theta \mid y_{s:1:t}) \times \prod_{m=1}^{s+K-2} q_{s+K-1, \theta}(x_{s+K-1}^m \mid x_{s+K}, y_{s+K-1:t}) \times \prod_{m=1}^{s+K-2} q_{s-1, \theta}(x_{s-1}^m \mid x_{s}^m, y_{s-1}) \times V_{s, \theta}^{k_s-1, \theta},
\]

(47)

and the marginal density of \( (x_{s}^{k_s}, \ldots, x_{s+K-1}^{k_s}, x_{s:M:t}, \theta) \) is \( \pi(x_{s}^{k_s}, \ldots, x_{s+K-1}^{k_s}, x_{s:M:t}, \theta \mid y_{s:t}) \).

**Proof** See Supplementary Material A.

Although the probability density (47) in Proposition 5.3 has a bit different form from that of (42) in Proposition 5.1, its marginal probability density is found to be the target posterior density \( \pi(x_{s:t}, \theta \mid y_{s:t}) \).

**Properties of the incremental weight.** Similar results to Proposition 5.2 hold for the backward block sampling, and are summarized in Proposition 5.4.

**Proposition 5.4** If

\[
(x_{s-1}^{k_s-1}, \ldots, x_{s+K-1}^{k_s-1}, x_{s+K:t}, k_s, k_s^*, \theta) \sim \frac{\pi(x_{s-1}^{k_s-1}, \ldots, x_{s+K-1}^{k_s-1}, x_{s+K:t}, \theta \mid y_{s-1:t})}{M^{K+1}}
\]

and

\[
(x_{s-1}^{-k_s-1}, \ldots, x_{s+K-1}^{-k_s-1}, a_s^{-k_s-1}, \ldots, a_{s+K-1}^{-k_s-1}, k_s^*) \sim \tilde{\psi}_\theta,
\]

\[ \odot \] Springer
where \( \hat{y}_t \) is given in (45), then

\[
E[\hat{p}(y_{s-1} \mid x_{s+K}, y_{s:t}, \theta)]^{-1} = E[\hat{p}(y_{s-1} \mid x_{s+K}, y_{s:t}, \theta)]^{-1} | x_{s+K}, y_{s:t}, \theta] = p(y_{s-1} \mid y_{s:t}, \theta)^{-1}.
\]

**Proof** See Supplementary Material A. \( \square \)

### 5.3 Particle simulation smoother

In Whiteley et al. (2010) and the discussion of Whiteley following Andrieu et al. (2010), the additional step is introduced to explore all possible ancestral lineages. This is expected to improve the mixing property of the particle Gibbs (see, e.g., Chopin and Singh (2015), Lee et al. (2020)). We also incorporate such a particle simulation smoother into the double block sampling based on the following proposition.

**Proposition 5.5** The joint conditional density of \((k_{i-K}^*, \ldots, k_t^*)\) is given by

\[
\hat{p}(k_{i-K}^*, \ldots, k_t^* | x_{s-1:t-K-1}, x_{t-K:t}^{1:M}, a_{t-K:t-1}, y_{s-1:t}, \theta) = \hat{p}(k_t^* | x_{s-1:t-K-1}, x_{t-K:t}^{1:M}, a_{t-K:t-1}, y_{s-1:t}, \theta) \times \prod_{t_0 = i}^{t-K} \hat{p}(k_{t_0}^* | x_{s-1:t-K-1}, x_{t-K:t_0}^{1:M}, a_{t-K:t_0-1}, x_{t_0+1}^{k_{t_0}^*}, \ldots, x_t^{k_t^*}, k_{t_0+1:t}^*, y_{s-1:t}, \theta),
\]

where

\[
\hat{p}(k_{t_0}^* | x_{s-1:t-K-1}, x_{t-K:t_0}^{1:M}, a_{t-K:t_0-1}, x_{t_0+1}^{k_{t_0}^*}, \ldots, x_t^{k_t^*}, k_{t_0+1:t}^*, y_{s-1:t}, \theta) = \tilde{V}_{t_0, \theta}^{k_{t_0}^*}, \quad \tilde{V}_{j, \theta}^m \equiv \frac{V_{j, \theta}^m f_\theta(x_{j+1}^m \mid x_j^m, y_{j+1})}{\sum_{i=1}^M V_{j, \theta}^i f_\theta(x_{j+1}^i \mid x_j^i, y_{j+1})}.
\]

**Proof** See Supplementary Material A. \( \square \)

Suppose we have \((x_{s-1:t-K-1}, x_{t-K:t}^{1:M}, a_{t-K:t-1}^1, k_t^*, \theta) \sim \hat{p}\) where \(\hat{p}\) is defined in (41). In Step 1-1(d), the lineage \(k_{t-K:t}^*\) is automatically determined when \(k_t^*\) is chosen. The particle simulation smoother breaks this relationship and again samples \(k_{t-K:t}^*\) jointly by generating \(k_{j}^* \sim \mathcal{N}(\tilde{V}_{j, \theta}^{1:M}, \theta = t-1, \ldots, t-K\), recursively.

### 6 Conclusion

In this paper, we propose a novel efficient estimation method to implement the rolling-window particle MCMC simulation using a SMC framework and refreshing steps with MCMC kernel. The weighted particles are updated to learn and discard the information of the new and old observations using the forward and backward block sampling based on the conditional SMC algorithm, which effectively circumvent the weight
degeneracy problem. The proposed estimation methodology is also applicable to the ordinary sequential estimation with parameter uncertainty. The illustrative examples show that our proposed sampler outperforms the simple rolling-window sampler.

**Supplementary Information**  The online version contains supplementary material available at https://doi.org/10.1007/s42081-022-00170-2.

**Acknowledgements**  We thank the Editor and two anonymous referees for their helpful comments. All computational results in this paper are generated using Ox metrics 7.0 (see Doornik (2009)). This work was supported by JSPS KAKENHI Grant numbers 25245035, 26245028, 17H00985, 15H01943, 19H00588.

**References**

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.

Chopin, N., Jacob, P. E., & Papaspiliopoulos, O. (2013). SMC2: An efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 75*(3), 397–426.

Chopin, N., & Singh, S. S. (2015). On particle Gibbs sampling. *Bernoulli, 21*(3), 1855–1883.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B: Statistical Methodology, 68*(3), 411–436.

Doornik, J. A. (2009). *An object-oriented matrix programming language—Ox 6*. London: Timberlake Consultants Press and Oxford. www.doornik.com.

Doucet, A., Briers, M., & Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics, 15*(3), 693–711.

Fulop, A., & Li, J. (2013). Efficient learning via simulation: A marginalized resample-move approach. *Journal of Econometrics, 176*(2), 146–161.

Heber, G., Lunde, A., Shephard, N., & Sheppard, K. (2009). Oxford-man Institute’s realized library. version 0.2, Oxford-Man Institute, University of Oxford.

Lee, A., Singh, S. S., & Vihola, M. (2020). Coupled conditional backward sampling particle filter. *Annals of Statistics, 48*(5), 3066–3089.

Naesseth, C.A., Lindsten, F., & Schön, T.B. (2015). Nested sequential Monte Carlo methods. arXiv preprint arXiv:1502.02536.

Omori, Y., Chib, S., Shephard, N., & Nakajima, J. (2007). Stochastic volatility with leverage: Fast and efficient likelihood inference. *Journal of Econometrics, 140*(2), 425–449.

Polson, N. G., Stroud, J. R., & Müller, P. (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society, Series B: Statistical Methodology, 70*(2), 413–428.

Shephard, N., & Sheppard, K. (2010). Realising the future: Forecasting with high-frequency-based volatility (HEAVY) models. *Journal of Applied Econometrics, 25*(2), 197–231.

Takahashi, M., Omori, Y., & Watanabe, T. (2009). Estimating stochastic volatility models using daily returns and realized volatility simultaneously. *Computational Statistics and Data Analysis, 53*(6), 2404–2426.

Whiteley, N., Andrieu, C., & Doucet, A. (2010). Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. arXiv preprint arXiv:1011.2437.

**Publisher’s Note**  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.