Foundations of Data Science
©American Institute of Mathematical Sciences
Volume **1**, Number **1**, March **2019**

doi:10.3934/fods.2019003

pp. **61–85**

PARTICLE FILTERS FOR INFERENCE OF HIGH-DIMENSIONAL MULTIVARIATE STOCHASTIC VOLATILITY MODELS WITH CROSS-LEVERAGE EFFECTS

Yaxian Xu and Ajay Jasra*

Department of Statistics & Applied Probability
National University of Singapore
Singapore, 117546, SG

Abstract. Multivariate stochastic volatility models are a popular and well-known class of models in the analysis of financial time series because of their abilities to capture the important stylized facts of financial returns data. We consider the problems of filtering distribution estimation and also marginal likelihood calculation for multivariate stochastic volatility models with cross-leverage effects in the high dimensional case, that is when the number of financial time series that we analyze simultaneously (denoted by d) is large. The standard particle filter has been widely used in the literature to solve these intractable inference problems. It has excellent performance in low to moderate dimensions, but collapses in the high dimensional case. In this article, two new and advanced particle filters proposed in [4], named the space-time particle filter and the marginal space-time particle filter, are explored for these estimation problems. The better performance in both the accuracy and stability for the two advanced particle filters are shown using simulation and empirical studies in comparison with the standard particle filter. In addition, Bayesian static model parameter estimation problem is considered with the advances in particle Markov chain Monte Carlo methods. The particle marginal Metropolis-Hastings algorithm is applied together with the likelihood estimates from the space-time particle filter to infer the static model parameter successfully when that using the likelihood estimates from the standard particle filter fails.

1. **Introduction.**

Univariate stochastic volatility (SV) models play a very important role in financial time series analysis and are widely used in areas such as economics and mathematical finance (see for example, [11, 18]). The reason of being so well-known and popular is the great flexibility that they provide in describing the stylized facts of financial time series. Univariate SV models are able to capture some famous stylized facts associated with financial data such as volatility clustering, the leverage effect and the fat-tailed nature of financial returns (see for example, [15, 18, 27]). As a result of its great power and flexibility, univariate SV modeling has become very useful and important in practice as well as theoretical studies.

2010 *Mathematics Subject Classification.* Primary: 82C80; Secondary: 60G99, 62F15.

Key words and phrases. Particle filters, multivariate stochastic volatility, filtering, high dimensions, particle Markov chain Monte Carlo.

AJ is supported by an AcRF tier 2 grant: R-155-000-161-112.

* Corresponding author.

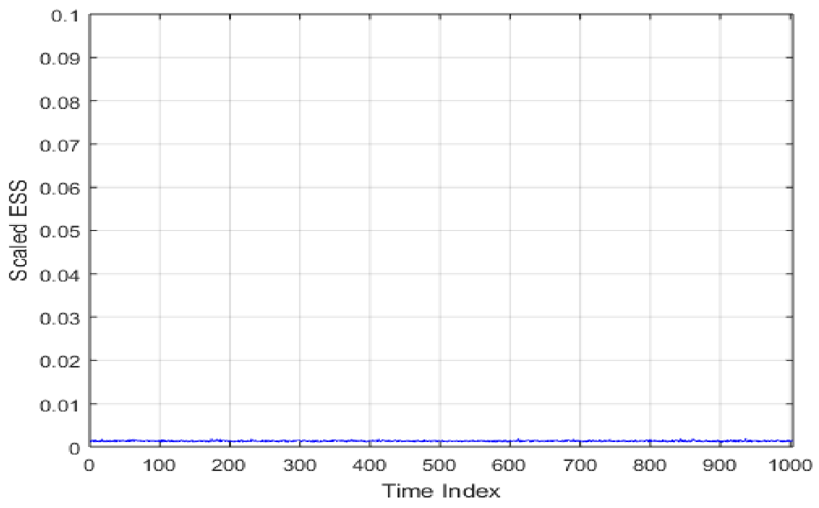
Multivariate stochastic volatility (MSV) models are natural extensions of univariate SV models and they are able to capture the covariation effect of financial returns in addition to all the advantages of univariate SV modeling. The covariation effect is another important stylized fact, which refers to the co-movement phenomenon of volatilities of different financial time series [27]. MSV modeling has therefore become a major concern to investigate the correlation structure of high-dimensional financial returns data [14] as it is able to model the movements in volatilities of a number of financial products simultaneously. MSV modeling takes advantage of more information and allows possible interactions between the volatilities of different time series, hence it is more comprehensive and normally leads to a better fit of the data as compared to univariate SV modeling [27]. There are extensive literatures investigating MSV models, see the surveys in [2] and [6] and the references therein for a more detailed introduction.

It is well-known that multivariate GARCH models provide an important alternative to MSV models for modeling financial volatility and have achieved many practical successes. For detailed survey of multivariate GARCH models, see [3]. These two classes of models have emerged as the dominant approaches to characterize volatilities that are inherent in financial time series data. However, as compared to GARCH-type models, MSV models are more flexible as a result of the introduction of random noise in the volatility process and they are more directly linked to continuous time models that are widely used in asset pricing in practice [2, 6]. The focus of this article is to extend the tool sets that can be used to analyze MSV models.

SV models normally involve two processes. An observed process $\{Y_t\}_{t\geq 1}$ representing the sequence of financial returns and a latent process $\{X_t\}_{t\geq 1}$ recording the evolvement of the volatilities. These two processes depend on some static model parameter θ. Of great interest is studying the distribution of the latent volatility at time t given the information provided by all the observed financial returns received up to that time point. This is known as the problem of filtering, i.e. estimating the filtering distributions $\{p_\theta(x_t|\, y_{1:t})\}_{t\geq 1}$. As making inference of the underlying volatilities from the observed sequences of returns is important, estimation of the filters has significant practical meanings. In addition, calculation of the marginal likelihood is normally a by-product of the particle approximation of the filters and it is important in areas such as model comparison and static parameter estimation [1, 12, 16, 18, 25]. Therefore, estimation of the filters and computation of the marginal likelihood for MSV models for a fixed and known θ are of our major concern. Moreover, we aim to estimate the model parameter θ as well. In particular, we consider the challenging situation when the number of financial time series that we analyze simultaneously (denoted by d) is large.

Since MSV models bring in non-linearity, there are no analytical solutions for the filters (see for example, [11, 20, 26]). As a result, we have to resort to numerical approximations to solve these intractable inference problems. There are a wide variety of efficient estimation methods for MSV models in the literature, see the reviews in [2, 6, 23, 24]. However, most of the approaches proposed are Markov chain Monte Carlo methods (see for example, [2, 14, 23]) and hence they are batch methods and applications are only found for low to moderate dimensional inference problems due to the computational complexity. Efficient on-line estimation methods for SV models are mentioned in [6, 18], but the high dimensionality issue has not been well investigated in the literature for the on-line inference of the filters. We aim

FIGURE 1. Plot of Scaled Effective Sample Size (ESS) averaged over 20 runs when standard particle filter is applied to MSV model with dimension 200



to confront the high dimensionality issue in this article with advanced sequential Monte Carlo (SMC) algorithms.

Standard particle filters are a popular class of algorithms that produce consistent estimators for the filters and facilitate the calculation of marginal likelihoods [11]. The standard particle filter generates a cloud of N weighted samples, termed particles, to approximate the distributions of interest. All these N particles are undergoing a sequence of mutations and selections sequentially in time. As a result of its outstanding performance in low to moderate dimensional problems, the standard particle filter is now widely used in areas as diverse as economics, finance, engineering and robotics [8, 11]. In particular, the standard particle filter has been a popular class of methods to solve these intractable inference problems for SV models (see for example, [11, 18]). However, when it is applied to high dimensional MSV models, the following Figure 1 shows that the standard particle filter collapses completely. The effective sample size, which is a measure of the variability of the importance weights, drops to almost 0 when we analyze 200 financial time series simultaneously. This indicates the collapse of weights and thus the failure of the standard particle filter. More details regarding this will be covered in section 3 and section 6.1.2.

Since high-dimensional MSV models are considered and the standard solution fails in this scenario, it would be interesting to investigate the performance of another two advanced particle filters dedicated to high-dimensional problems, named the space-time particle filter (STPF) and the marginal STPF proposed in [4]. The STPF is designed for a specific family of state-space models satisfying some factorization structure detailed in section 4.1 and works well when there is a spatial mixing element in the dimension of the hidden state. It confronts the high dimensionality issue by introducing additional intermediate SMC steps. The STPF builds up a local particle filter that moves vertically along the space direction and a global particle filter that moves horizontally along the time steps [22, 4]. It provides consistent estimators for the filters and more stable results as compared with the standard particle filter since it scales much better in the high-dimensional cases, at least in some examples. The marginal STPF combines the ideas of marginal particle filter (MPF) and resample-move algorithm within the STPF framework. As suggested by the numerical studies in [4], the marginal STPF algorithm has excellent performance in high-dimensional problems. However, it incurs substantially higher computational cost.

The contributions of this article are as follows. First, the tool set for the analysis of MSV models is extended with our study. In this article, we develop the formal framework for the applications of the standard particle filter, the STPF and the marginal STPF to high-dimensional MSV models for the problems of filtering distribution estimation and marginal likelihood calculation and compare their performance. Supported by the results of both the simulation and empirical studies, we provide users with other alternatives (they are the STPF and the marginal STPF) for the above-mentioned estimation problems when the standard particle filter fails. Second, the particle marginal Metropolis-Hastings algorithm is applied together with the likelihood estimates from the STPF to estimate the static model parameter successfully while that using the likelihood estimates from the standard particle filter is not able to do so. In addition, since only simulation studies are conducted in [4], whether these two new algorithms proposed are applicable to real life examples remains unknown. Through our efforts, we have verified the validity of these two new algorithms using real life financial data.

The organization of the rest of this article is as follows. We provide a detailed introduction of MSV models in section 2. A comprehensive literature review of the standard particle filter and its application to MSV models with cross-leverage effects are detailed in section 3. In section 4, we introduce the STPF and the marginal STPF and how these algorithms can be applied to MSV models with cross-leverage effects. How the results in section 3 and 4 can be used to infer the static model parameter is discussed in section 5. Simulation and empirical studies are presented in section 6 and we finish this article with a conclusion in section 7. For simplicity, we suppress the static model parameter θ from the notations in section 3 and 4 since in these two sections, θ is assumed to be fixed and known.

2. **Multivariate stochastic volatility model.**

2.1. **Univariate stochastic volatility model.**

Univariate SV models are well-known and popular in the analysis of financial data as they are able to model the time-varying variances of financial returns [2, 6, 11, 27]. This class of models has received a lot of attention in the econometrics literature as it is a way to generalize the Black-Scholes formula for option pricing to allow stochastic volatility [13, 26]. Let Y_t denote the stock return at time t and X_t be the latent log-volatility at time t. The univariate SV model (see [14, 18, 26] for details) is given by

$$Y_t = e^{(X_t/2)}\varepsilon_t, \;\; t = 1, ..., n,$$

$$X_{t+1} = \phi X_t + \eta_t, \;\; t = 1, ..., n-1,$$

where

$$\begin{pmatrix}\varepsilon_t \\ \eta_t\end{pmatrix} \sim \mathbf{N}_2\left(\mathbf{0}, \Sigma\right), \;\; \Sigma = \begin{pmatrix}\sigma_\varepsilon^2 & \rho\sigma_\varepsilon\sigma_\eta \\ \rho\sigma_\varepsilon\sigma_\eta & \sigma_\eta^2\end{pmatrix}, \tag{1}$$

and $\mathbf{N}_d(\boldsymbol{\mu}, \Sigma)$ denotes a d-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ.

In addition, the initial distribution of the latent log-volatility is given by

$$X_1 \sim \mathbf{N}\left(0, \frac{\sigma_\eta^2}{1-\phi^2}\right), \tag{2}$$

so that the marginal distributions of all latent variables X_t $(t = 2, ..., n)$ are the same as the normal distribution given by (2).

Here, ϕ can be regarded as the persistence in the volatility and σ_η^2 is the volatility of the volatility [18, 26]. Univariate SV model with leverage effect is considered if ρ in (1) is nonzero. A nonzero ρ indicates the correlation between the stock return at time t and the future log-volatility at time $t+1$. This type of model is able to capture the leverage effect that leads to the asymmetry phenomenon of volatilities. To be more specific, negative returns normally produce a rise in volatility and positive returns normally lead to a fall in volatility [2, 27].

2.2. Multivariate stochastic volatility model.

MSV models are natural extensions of univariate SV models in which we consider d univariate SV models simultaneously. Here d can be understood to be the number of stocks that are under study. With some abuse of notations, let $Y_t = \left(Y_{1t}, \cdots, Y_{dt}\right)'$ denote a d-dimensional stock return vector and $X_t = \left(X_{1t}, \cdots, X_{dt}\right)'$ denote a d-dimensional log-volatility vector, MSV model (see [2, 6, 14] for details) is given by

$$Y_t = V_t^{1/2}\varepsilon_t, \;\; t = 1, ..., n, \tag{3}$$

$$X_{t+1} = \Phi X_t + \eta_t, \;\; t = 1, ..., n-1, \tag{4}$$

where

$$V_t = \begin{pmatrix} e^{X_{1t}} & & \\ & \ddots & \\ & & e^{X_{dt}} \end{pmatrix},$$

$$\Phi = \begin{pmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_d \end{pmatrix},$$

and

$$\begin{pmatrix} \varepsilon_t \\ \eta_t \end{pmatrix} \sim \mathbf{N}_{2d}(\mathbf{0}, \Sigma), \;\; \Sigma = \begin{pmatrix} \Sigma_{\varepsilon\varepsilon} & \Sigma_{\varepsilon\eta} \\ \Sigma_{\eta\varepsilon} & \Sigma_{\eta\eta} \end{pmatrix}. \tag{5}$$

The initial distribution of the log-volatility vector X_1 is given by

$$X_1 \sim \mathbf{N}_d(\mathbf{0}, \Sigma_0),$$

and the $(i,j)^{th}$ entry of Σ_0 is given by the quotient of the $(i,j)^{th}$ entry of $\Sigma_{\eta\eta}$ and $1-\phi_i\phi_j$ so that the marginal distributions of all log-volatility vectors X_t $(t = 1, ..., n)$ are the same.

Nonzero diagonal and off-diagonal entries of $\Sigma_{\varepsilon\eta}$ indicate the existence of leverage and cross-leverage effects respectively (for details see [2, 14]). A nonzero $(i,j)^{th}$ entry in $\Sigma_{\varepsilon\eta}$ shows the correlation between the i^{th} stock return at time t and the future volatility for stock j at time $t+1$. Therefore, cross-leverage effects allow possible interactions between different financial time series.

Since MSV model with cross-leverage effects shares great advantage of its flexibility to capture the important stylized facts of financial data, such as time-varying volatilities, the leverage effect and the covariation effect, the model of interest in this article is the high-dimensional MSV model with cross-leverage effects.

2.3. **Problems of interest.** Our goal is to make inference about the hidden log-volatilities $\{X_t\}_{t\geq 1}$ while we only have access to the observed stock returns $\{Y_t\}_{t\geq 1}$ for a fixed and known model parameter $\theta = (\Phi, \Sigma)$. In particular, in the situation when a large d is considered. To be more specific, first we aim to estimate the filtering distributions $\{p_\theta(x_t|\, y_{1:t})\}_{t\geq 1}$, which is the distribution of the latent volatility at time t given the information provided by all the observed financial returns received up to that time point. This is termed the problem of filtering, which is an important problem in econometrics and mathematical finance [11]. Second we aim to compute the marginal likelihood $\{p_\theta(y_{1:t})\}_{t\geq 1}$, which can be computed recursively by

$$p_\theta(y_{1:t}) = p_\theta(y_1) \prod_{k=2}^{t} p_\theta(y_k|\, y_{1:k-1}). \tag{6}$$

This quantity is useful for model comparisons as it is required in the computation of likelihood ratio statistics and Bayes factor [18]. In addition, it is rather useful for static parameter estimation when the true likelihoods cannot be solved analytically [1, 12, 16, 25]. As a result, making inference of the model parameter θ is the third goal of this article.

3. **Standard particle filter and its application.** Taking into account of the leverage and cross-leverage effects of the MSV models introduced in section 2.2, $\left(Y_t,\, X_{t+1} \,|\, X_t\right)'$ are jointly distributed. More precisely, from (3) and (4), we can derive that

$$\left(\begin{array}{c|c} Y_t & \\ & X_t \\ X_{t+1} & \end{array} \right) \sim \mathbf{N}_{2d}\left(\begin{pmatrix} \mathbf{0} \\ \Phi X_t \end{pmatrix}, \begin{pmatrix} V_t^{1/2}\Sigma_{\varepsilon\varepsilon}V_t^{1/2} & V_t^{1/2}\Sigma_{\varepsilon\eta} \\ \Sigma_{\eta\varepsilon}V_t^{1/2} & \Sigma_{\eta\eta} \end{pmatrix} \right). \tag{7}$$

Since this joint distribution can be decomposed into

$$f\left(y_t,\, x_{t+1}|x_t\right) = f\left(x_{t+1}|\, x_t, y_t\right) g\left(y_t|\, x_t\right),$$

the evolution of the hidden volatility from X_{t-1} to X_t can be thought of as a stochastic process that depends on its previous state X_{t-1} and previous observed return Y_{t-1}. From now on, we denote $f\left(x_{t+1}|\, x_t, y_t\right)$ to be the transition density of the hidden volatility and $g\left(y_t|\, x_t\right)$ to be the measurement density for MSV model with cross-leverage effects given by (3) and (4). Let $\mu\left(x_1\right)$ represent the initial distribution of the log-volatility X_1. We can see that MSV model with cross-leverage effects fits in the framework of general state-space models and therefore, the particle methods for state-space models can be applied.

There are no analytical solutions to the sequence of filtering distributions $\{p\left(x_t|\, y_{1:t}\right)\}_{t\geq 1}$ in this case. Thus, we have to resort to numerical solutions. One widely used numerical approximation method is Sequential Monte Carlo (SMC) method, which is also known as the Particle Filter (PF). We will term it the standard PF from now onwards. The standard PF relies on the numerical approximations of $\{p\left(x_{1:t}|\, y_{1:t}\right)\}_{t\geq 1}$ rather than directly on the filters $\{p\left(x_t|\, y_{1:t}\right)\}_{t\geq 1}$. If we are able to approximate $\{p\left(x_{1:t}|\, y_{1:t}\right)\}_{t\geq 1}$ sequentially in time, the filters $\{p\left(x_t|\, y_{1:t}\right)\}_{t\geq 1}$ will be obtained easily by discarding the samples for $X_1, ..., X_{t-1}$. Therefore, $\{p\left(x_{1:t}|\, y_{1:t}\right)\}_{t\geq 1}$ is the sequence of target distributions that we are interested in, with $\{p\left(x_{1:t}, y_{1:t}\right)\}_{t\geq 1}$ being the sequence of unnormalized targets and $\{p\left(y_{1:t}\right)\}_{t\geq 1}$ being the corresponding normalizing constants.

It is not hard to check that the unnormalized posterior distribution $p\left(x_{1:t}, y_{1:t}\right)$ for MSV models with cross-leverage effects satisfies the following recursive relation

$$p\left(x_{1:t}, y_{1:t}\right) = p\left(x_{1:t-1}, y_{1:t-1}\right) f\left(x_t \middle| x_{t-1}, y_{t-1}\right) g\left(y_t \middle| x_t\right). \tag{8}$$

Thus, the posterior distribution $p\left(x_{1:t} \middle| y_{1:t}\right)$ satisfies

$$p\left(x_{1:t} \middle| y_{1:t}\right) = p\left(x_{1:t-1} \middle| y_{1:t-1}\right) \frac{f\left(x_t \middle| x_{t-1}, y_{t-1}\right) g\left(y_t \middle| x_t\right)}{p\left(y_t \middle| y_{1:t-1}\right)}, \tag{9}$$

where

$$p\left(y_t \middle| y_{1:t-1}\right) = \int p\left(x_{t-1} \middle| y_{1:t-1}\right) f\left(x_t \middle| x_{t-1}, y_{t-1}\right) g\left(y_t \middle| x_t\right) \mathrm{d}x_{t-1:t}. \tag{10}$$

Since it is difficult to sample directly from our target density $p\left(x_{1:t} \middle| y_{1:t}\right)$, importance sampling (IS) is used in the standard PF. IS is a well-known sampling tool that utilizes a proposal density and a weighting procedure to provide a collection of weighted samples that approximates the target. In particular, we are interested in a sequence of target densities of increasing dimensions $\{p\left(x_{1:t} \middle| y_{1:t}\right)\}_{t\geq 1}$. In order to avoid the increasing computational burden that grows with t, sequential importance sampling (SIS) is used. SIS is a special instance of IS in which the proposal density admits the following recursive structure

$$\begin{aligned} q_t\left(x_{1:t}\right) &= q_{t-1}\left(x_{1:t-1}\right) q_t\left(x_t \middle| x_{1:t-1}\right) \\ &= q_1\left(x_1\right) \prod_{k=2}^{t} q_k\left(x_k \middle| x_{1:k-1}\right). \end{aligned} \tag{11}$$

This structure of the proposal density results in the decomposition of the unnormalized weight function in the following recursive way

$$\begin{aligned} w_t\left(x_{1:t}\right) &= \frac{p\left(x_{1:t}, y_{1:t}\right)}{q_t\left(x_{1:t}\right)} \\ &= \frac{p\left(x_{1:t-1}, y_{1:t-1}\right)}{q_{t-1}\left(x_{1:t-1}\right)} \frac{p\left(x_{1:t}, y_{1:t}\right)}{p\left(x_{1:t-1}, y_{1:t-1}\right) q_t\left(x_t \middle| x_{1:t-1}\right)} \\ &= w_{t-1}\left(x_{1:t-1}\right) \alpha_t\left(x_{1:t}\right) \end{aligned} \tag{12}$$

where $\alpha_t\left(x_{1:t}\right)$ is the incremental importance weight function given by

$$\begin{aligned} \alpha_t\left(x_{1:t}\right) &= \frac{p\left(x_{1:t}, y_{1:t}\right)}{p\left(x_{1:t-1}, y_{1:t-1}\right) q_t\left(x_t \middle| x_{1:t-1}\right)} \\ &= \frac{f\left(x_t \middle| x_{t-1}, y_{t-1}\right) g\left(y_t \middle| x_t\right)}{q_t\left(x_t \middle| x_{1:t-1}\right)}. \end{aligned} \tag{13}$$

For a detailed introduction of IS and SIS, please refer to [11]. According to SIS, at time t, we have a cloud of weighted samples $\{W_t^i, X_{1:t}^i\}(i = 1, ..., N)$ for the approximation of the target density. Here N is the number of samples generated and W_t^i denotes the normalized weight for the i^{th} particle, which is obtained by

$$W_t^i = \frac{w_t\left(X_{1:t}^i\right)}{\sum_{j=1}^{N} w_t\left(X_{1:t}^j\right)}.$$

However, there exists a severe drawback with IS and SIS that the variance of the weights $\{W_t^i\}(i = 1, ..., N)$ increases with the time parameter t in the majority of the cases [4, 5, 10, 21]. Remedy to this drawback involves an additional resampling procedure in the SIS algorithm. The idea of resampling is quite intuitive.

We sample, with replacement, N particles from the current cloud of weighted samples $\{W_t^i, X_{1:t}^i\}$ and reset the weight to $\frac{1}{N}$. Through this procedure, we are able to remove particles with low weights and multiply particles with high weights so that we don't need to carry forward particles with vanishingly small weights. The most popular three resampling schemes include multinomial resampling, residual resampling and systematic resampling [19].

We should not resample at every time step as it can be very inefficient and introduce some additional variance into the estimation procedure [7, 11]. Therefore, the idea of dynamic resampling should be used (see [9] and the references therein). It is more reasonable to resample only when the variance of the weights moves beyond a pre-selected threshold. This is done by evaluating the effective sample size (ESS) measure, which is given by

$$ESS = \frac{1}{\sum_{i=1}^N (W_t^i)^2}. \tag{14}$$

ESS is a number between 1 and N and it indicates roughly the number of useful samples that the algorithm has. Resampling is needed only when ESS falls below some threshold N_T, typically $N_T = \frac{N}{2}$.

In summary, the standard PF is a combination of SIS and resampling. It provides approximations to a sequence of related distributions of increasing dimensions. At each step t, the target is approximated by a cloud of N weighted samples termed particles. Assume that $\{W_t^i, X_{1:t}^i\}(i = 1, ..., N)$ and $\{\overline{W}_t^i, \overline{X}_{1:t}^i\}(i = 1, ..., N)$ denotes the cloud of weighted particles at time index t before and after the resampling step respectively, $p\left(x_t \middle| y_{1:t}\right)$ and $p\left(y_t \middle| y_{1:t-1}\right)$ can be approximated as follows.

$$\widehat{p}\left(x_t \middle| y_{1:t}\right) = \sum_{i=1}^N W_t^i \, \delta_{X_t^i}\left(x_t\right),$$

or

$$\widehat{p}\left(x_t \middle| y_{1:t}\right) = \sum_{i=1}^N \overline{W}_t^i \, \delta_{\overline{X}_t^i}\left(x_t\right),$$

where $\delta_{X_t^i}\left(x_t\right)$ denotes the Dirac delta mass located at X_t^i.

$$\widehat{p}\left(y_t \middle| y_{1:t-1}\right) = \sum_{i=1}^N \overline{W}_{t-1}^i \, \alpha_t\left(X_{1:t}^i\right).$$

For MSV models with cross-leverage effects, if $f\left(x_t \middle| x_{t-1}, y_{t-1}\right)$ is chosen to be the proposal density $q_t\left(x_t \middle| x_{1:t-1}\right)$, then $\alpha_t\left(x_{1:t}\right) = g\left(y_t \middle| x_t\right)$. Algorithm 1 details the standard PF for MSV models with cross-leverage effects. This algorithm is run sequentially in time for each value of t. For notational convenience, we set $1:0 = \emptyset$, $\overline{W}_0^i = \frac{1}{N}$ and $\widehat{p}\left(y_{1:0}\right) = 1$.

Both $f\left(x_t \middle| x_{t-1}, y_{t-1}\right)$ and $g\left(y_t \middle| x_t\right)$ can be obtained easily from (7), with

$$X_t | X_{t-1}, Y_{t-1} \sim \mathbf{N}_d\left(\Phi X_{t-1} + \Sigma_{\eta\varepsilon}\Sigma_{\varepsilon\varepsilon}^{-1} V_{t-1}^{-1/2} Y_{t-1},\, \Sigma_{\eta\eta} - \Sigma_{\eta\varepsilon}\Sigma_{\varepsilon\varepsilon}^{-1}\Sigma_{\varepsilon\eta}\right), \tag{15}$$

and

$$Y_t | X_t \sim \mathbf{N}_d\left(\mathbf{0},\, V_t^{1/2}\Sigma_{\varepsilon\varepsilon}V_t^{1/2}\right). \tag{16}$$

As a result of the resampling step, looking back in time, many of the particles will be exactly the same. This is referred to as the path degeneracy problem (see for example, [11, 20] for details), which is the major limitation of the standard PF. For example, any approximation that relies on the full path $x_{1:t}$ will fail if the standard

Algorithm 1 Standard PF for MSV model with cross-leverage effects

```
1:  for $i \in \{1, \dots, N\}$ do
2:      if $t = 1$ then
3:          Sample $X_t^i \sim \mu(x_1)$
4:      else
5:          Sample $X_t^i \sim f(x_t | \overline{X}_{t-1}^i, y_{t-1})$
6:      end if
7:      Set $X_{1:t}^i \leftarrow (\overline{X}_{1:t-1}^i, X_t^i)$
8:      Compute $\alpha_t(X_{1:t}^i) = g(y_t | X_t^i)$
9:      Update $w_t(X_{1:t}^i) = \overline{W}_{t-1}^i \alpha_t(X_{1:t}^i)$
10: end for
11: Calculate $\widehat{p}(y_{1:t}) = \widehat{p}(y_{1:t-1}) \sum_{i=1}^N w_t(X_{1:t}^i)$
12: Normalize $W_t^i \propto w_t(X_{1:t}^i)$
13: Calculate ESS and resample if necessary
14: Set $\{\overline{W}_t^i, \overline{X}_{1:t}^i\} \leftarrow \{\frac{1}{N}, \overline{X}_{1:t}^i\}$ (with resampling) or $\{\overline{W}_t^i, \overline{X}_{1:t}^i\} \leftarrow \{W_t^i, X_{1:t}^i\}$
    (without resampling)
```

PF is performed for sufficiently many time steps since every resampling step reduces the number of unique particles representing X_1. This path degeneracy problem will be addressed in the next section.

4. The Space-Time particle filter.

4.1. The Space-Time particle filter and its application.

The standard PF performs quite well in inference problems for general state-space models with low to moderate dimensions [8]. However, the number of particles required scales exponentially with the model dimension d for a successful application of the standard PF. High-dimensional standard PFs will collapse with almost all the weight assigned to a single particle while all the other particles have vanishingly small weights if N is not large enough. Therefore, for the algorithm to be stable, the standard PF incurs a cost that is exponential in d [4, 5, 28].

As a result of the high-dimensional nature of the problem that we are considering, it is interesting to explore some advanced SMC methods within our context. The STPF has been developed recently and is dedicated to high-dimensional problems [4]. It provides consistent Monte Carlo estimates for any fixed d as N grows. Moreover, the STPF provides more stable estimates as it scales much better with large d than the standard PF, at least for some examples, as shown by the empirical results provided in [4]. We will give a brief introduction of the STPF and then apply it to MSV models with cross-leverage effects.

According to [4], the STPF is designed for a specific family of HMMs satisfying the following factorization structure (17). Let $f(x_t | x_{t-1})$ and $g(y_t | x_t)$ denote the transition density and measurement density of a general state-space model.

$$f(x_t | x_{t-1})\, g(y_t | x_t) = \prod_{j=1}^{T_{t,d}} \alpha_{t,j}(y_t, x_{t-1}, x_t(A_{t,j})), \tag{17}$$

where $\{A_{t,j}\}_{j=1}^{T_{t,d}}$ is an increasing sequence of sets with $A_{t,1} \subset A_{t,2} \subset \dots \subset A_{t,T_{t,d}} = \{1 : d\}$ for some integer $0 < T_{t,d} \leq d$, $\alpha_{t,j}$ are some properly chosen functions and $x_t(A) = \{x_t(j) : j \in A\}$. A common choice for $A_{t,j}$ is $\{1 : j\}$ and $T_{t,d} = d$. The most important factor for this algorithm to be effective is that this factorization needs to allow a gradual introduction of the full likelihood $g\left(y_t|\, x_t\right)$ along the $T_{t,d}$ steps.

The STPF can be thought of as augmenting the sequence of distributions moving from $p\left(x_{1:t-1}|\, y_{1:t-1}\right)$ to $p\left(x_{1:t}|\, y_{1:t}\right)$ with some intermediate distributions. In other words, the factorization structure defined in (17) will be utilized to build a PF that moves along the space direction. The idea of PF within PF is used in the STPF, and is motivated by the bootstrap within bootstrap island filter described in [29]. To be more specific, the t^{th} time step of the PF is broken down into $T_{t,d}$ space steps and a local PF with M_d particles is run along the space steps [4]. As a result of this break-down of the problem, IS in a single step for a d dimensional object is now turned into $T_{t,d}$ IS steps for objects of much lower dimensions. Since the standard PF normally performs quite well in low to moderate dimensional problems, we expect the STPF to work well even in the high-dimensional cases. When there is a spatial mixing element in the dimension of the hidden state, the STPF exhibits certain stability properties at a sub-exponential cost that is polynomial in d.

The implementation of the algorithm combines a global filter with N particles running horizontally along the time steps, with a local filter with M_d particles moving vertically along the space steps. A motivation of adopting N global filters simultaneously instead of only 1 global filter is to deal with the path degeneracy problem of PFs if d is large. In the case of a large d, even within a single time step, the M_d local particles can exhibit path degeneracy. Thus, in order to provide reliable estimates for expectations over the complete d dimensions (for example, the mean of the filters), N global particle systems will be considered simultaneously. For a detailed description of STPF, please see [4].

Following the idea introduced above, the factorization structure specified by (17) is satisfied by MSV models with cross-leverage effects as follows

$$f\left(x_t|\, x_{t-1}, y_{t-1}\right) g\left(y_t|\, x_t\right) = \prod_{j=1}^{d}\left(p\left(x_t(j)|\, x_{t-1}, y_{t-1}, x_t(1:j-1)\right) \frac{p\left(y_t(1:j)|\, x_t(1:j)\right)}{p\left(y_t(1:j-1)|\, x_t(1:j-1)\right)}\right), \tag{18}$$

with $p\left(y_t(1:0)|\, x_t(1:0)\right) := 1$ and thus

$$\begin{aligned}&\alpha_{t,j}\left(y_t, x_{t-1}, x_t(1:j)\right)\\ &= p\left(x_t(j)|\, x_{t-1}, y_{t-1}, x_t(1:j-1)\right) \frac{p\left(y_t(1:j)|\, x_t(1:j)\right)}{p\left(y_t(1:j-1)|\, x_t(1:j-1)\right)}.\end{aligned} \tag{19}$$

At the local level, if $p\left(x_t(j)|\, x_{t-1}, y_{t-1}, x_t(1:j-1)\right)$ is selected to be the proposal, then $\frac{p\left(y_t(1:j)|\, x_t(1:j)\right)}{p\left(y_t(1:j-1)|\, x_t(1:j-1)\right)}$ will be the incremental weight function. As a result of (15) and (16), $p\left(x_t(j)|\, x_{t-1}, y_{t-1}, x_t(1:j-1)\right)$ and $p\left(y_t(1:j)|\, x_t(1:j)\right)$ can be derived easily. Following the detailed algorithm outlined in [4], the pseudocode of STPF for MSV model with leverage and cross-leverage effects proceeds as below in Algorithm 2. This algorithm is run sequentially in time for each value of t.

Here, for $X_t^{i,l}(1:j)$, $i \in \{1, ..., N\}$ denotes the i^{th} global particle, $l \in \{1, ..., M_d\}$ denotes the l^{th} local particle, $t \geq 1$ denotes the discrete observation time and $1:j$ denotes the space steps $1, ..., j$. For simplicity, we assume that resampling is conducted at the end of each time and space step. We set $1:0 = \emptyset$, $\widehat{p}\left(y_{1:0}\right) = 1$, $\check{x}_{0,j-1}^{i,l}(1:d) = \mathbf{0}$ and $\check{x}_{t-1,0}^{i,l}(1:d) = \bar{x}_{t-1}^{i,l}(1:d)$.

Algorithm 2 Space-Time Particle Filter for MSV model with cross-leverage effects

```
1: for $i \in \{1, \ldots, N\}$ do
2:     for $j \in \{1, \ldots, d\}$ do
3:         for $l \in \{1, \ldots, M_d\}$ do
4:             if $t = 1$ then
5:                 Sample $x_t^{i,l}(j) \sim p\left(x_1(j) \middle| \breve{x}_1^{i,l}(1:j-1)\right)$
6:             else
7:                 Sample $x_t^{i,l}(j) \sim p\left(x_t(j) \middle| \breve{x}_{t-1,j-1}^{i,l}(1:d), y_{t-1}, \breve{x}_t^{i,l}(1:j-1)\right)$
8:             end if
9:             Set $x_t^{i,l}(1:j) \leftarrow \left(\breve{x}_t^{i,l}(1:j-1), x_t^{i,l}(j)\right)$
10:            Calculate $G_{t,j}^{i,l} = \frac{p\left(y_t(1:j) \middle| x_t^{i,l}(1:j)\right)}{p\left(y_t(1:j-1) \middle| \breve{x}_t^{i,l}(1:j-1)\right)}$
11:        end for
12:        Normalize $W_{t,j}^{i,l} \propto G_{t,j}^{i,l}$
13:        Resample $\{x_t^{i,l}(1:j)\}_{l=1}^{M_d}$, together with $\{\breve{x}_{t-1,j-1}^{i,l}(1:d)\}_{l=1}^{M_d}$, according
       to $\{W_{t,j}^{i,l}\}_{l=1}^{M_d}$ and denote $\{\breve{x}_t^{i,l}(1:j)\}_{l=1}^{M_d}$ and $\{\breve{x}_{t-1,j}^{i,l}(1:d)\}_{l=1}^{M_d}$ to be the equally
       weighted resampled particles
14:    end for
15:    Calculate $\mathbf{G}_t^i = \prod_{j=1}^{d} \left(\frac{1}{M_d} \sum_{l=1}^{M_d} G_{t,j}^{i,l}\right)$, the weight associated with the $i^{th}$
       particle system
16: end for
17: Calculate $\widehat{p}(y_{1:t}) = \widehat{p}(y_{1:t-1})\left(\frac{1}{N} \sum_{i=1}^{N} \mathbf{G}_t^i\right)$
18: Normalize $\mathbf{W}_t^i \propto \mathbf{G}_t^i$
19: Resample $\{\mathbf{W}_t^i, \breve{x}_t^{i,1:M_d}(1:d)\}_{i=1}^{N}$ to obtain $N$ equally weighted particle systems
    $\{\frac{1}{N}, \bar{x}_t^{i,1:M_d}(1:d)\}_{i=1}^{N}$
```

4.2. **The marginal Space-Time particle filter and its application.** Even though the STPF is expected to have a better performance than the standard PF in the high-dimensional situations, path degeneracy problem that arises if d is overly large may still limit its success [4]. The marginal STPF algorithm is a further improvement based upon the STPF and is suited for high-dimensional cases to deal with this degeneracy problem. It combines the ideas of resample-move algorithm and the marginal PF (MPF) within the STPF framework. Resample-move algorithm incorporates Markov Chain Monte Carlo (MCMC) moves after the resampling step in a PF to achieve greater particle diversity and thus mitigate the path degeneracy problem (for details, see [11]). It is a simple idea that can be used in any SMC algorithm at the expense of increased computational efforts. The MPF is introduced in [20]. It operates directly on the marginal space of fixed dimensions by performing a marginal IS on the filter $p(x_t | y_{1:t})$. As a result of performing IS on a much lower dimensional space than the standard PF, a significant reduction in the variance of importance weights is expected for the MPF. However, this comes at a computational cost.

Recall that the STPF runs N global particle systems simultaneously to deal with the path degeneracy problem for large values of d. In contrast, the marginal STPF algorithm can simply be applied with only 1 global particle system (i.e. $N = 1$) to address the same issue. Below are the details how marginal STPF can be applied to MSV model with leverage and cross-leverage effects.

At time step 1, resample-move algorithm is utilized to improve particle diversity at each space step for each local particle. This involves the insertion of MCMC moves using some Markov transition kernel $K_{1,j}\left(x_1(1:j)\middle|\, x_1'(1:j)\right)$ with stationary distribution proportional to

$$\prod_{k=1}^{j} \alpha_{1,k}\left(y_1, x_0, x_1(1:k)\right) \tag{20}$$

after resampling in the j^{th} space step.

For MSV models with cross-leverage effects, (20) is reduced to

$$\prod_{k=1}^{j}\left(p\left(x_1(k)\middle|\, x_1(1:k-1)\right)\frac{p\left(y_1(1:k)\middle|\, x_1(1:k)\right)}{p\left(y_1(1:k-1)\middle|\, x_1(1:k-1)\right)}\right) =$$
$$p\left(x_1(1:j)\right)p\left(y_1(1:j)\middle|\, x_1(1:j)\right).$$

At time step $t \geq 2$, the ideas of the MPF and resample move are combined. Since only the filtering distribution $p\left(x_t\middle|\, y_{1:t}\right)$ is desired, the idea of MPF can be used here so that we only operate on the x_t-space, but not on the joint (x_{t-1}, x_t)-space by marginalizing over x_{t-1} [4]. Specifically, the target density for the MPF is proportional to

$$\sum_{l=1}^{M_d}\prod_{k=1}^{j} \alpha_{t,k}\left(y_t, \bar{x}_{t-1}^{i,l}(1:d), x_t(1:k)\right) \tag{21}$$

for the i^{th} particle system at the j^{th} space step. Similar to time step 1, MCMC moves are then incorporated using some Markov transition kernel $K_{t,j}\left(x_t(1:j)\middle|\, x_t'(1:j)\right)$ with the above stationary distribution.

For MSV models with cross-leverage effects, (21) is reduced to

$$\sum_{l=1}^{M_d}\prod_{k=1}^{j}\left(p\left(x_t(k)\middle|\, \bar{x}_{t-1}^{i,l}(1:d), y_{t-1}, x_t(1:k-1)\right)\frac{p\left(y_t(1:k)\middle|\, x_t(1:k)\right)}{p\left(y_t(1:k-1)\middle|\, x_t(1:k-1)\right)}\right)$$
$$= p\left(y_t(1:j)\middle|\, x_t(1:j)\right)\sum_{l=1}^{M_d} p\left(x_t(1:j)\middle|\, \bar{x}_{t-1}^{i,l}(1:d), y_{t-1}\right) \tag{22}$$

For $t \geq 2$, if the proposal density at time t and space step j is denoted by $q_{t,j}\left(x_t(j)\right)$, then the weight function at space step 1 for the i^{th} particle system is

$$\frac{p\left(y_t(1)\middle|\, x_t(1)\right)\sum_{m=1}^{M_d} p\left(x_t(1)\middle|\, \bar{x}_{t-1}^{i,m}(1:d), y_{t-1}\right)}{q_{t,1}\left(x_t(1)\right)} \tag{23}$$

and the incremental weight function for space step $j \geq 2$ is reduced to

$$\frac{p\left(y_t(1:j)\middle|\, x_t(1:j)\right)\sum_{m=1}^{M_d} p\left(x_t(1:j)\middle|\, \bar{x}_{t-1}^{i,m}(1:d), y_{t-1}\right)}{p\left(y_t(1:j-1)\middle|\, x_t(1:j-1)\right)\sum_{m=1}^{M_d} p\left(x_t(1:j-1)\middle|\, \bar{x}_{t-1}^{i,m}(1:d), y_{t-1}\right)q_{t,j}\left(x_t(j)\right)} \tag{24}$$

As long as the MCMC moves are implemented effectively, the marginal STPF algorithm should be able to mitigate the degeneracy problem, at the expense of increased computational load. The pseudocode for the marginal STPF algorithm for MSV model with leverage and cross-leverage effects (with resampling at every space and time step) is written as below in Algorithm 3. The proposals we used in our numerical studies for the marginal STPF are the same as those in the STPF.

Algorithm 3 Marginal Space-Time Particle Filter for MSV model with cross-leverage effects

```
1: At time step $t = 1$:
2: for $i \in \{1, \dots, N\}$ do
3:     for $j \in \{1, \dots, d\}$ do
4:         for $l \in \{1, \dots, M_d\}$ do
5:             Sample $x_1^{i,l}(j) \sim q_{1,j}\left(x_1(j)\right)$
6:             Set $x_1^{i,l}(1:j) \leftarrow \left(\tilde{x}_1^{i,l}(1:j-1),\, x_1^{i,l}(j)\right)$
7:             Calculate $G_{1,j}^{i,l} = \frac{p\left(x_1^{i,l}(1:j)\right) p\left(y_1(1:j) \mid x_1^{i,l}(1:j)\right)}{p\left(\tilde{x}_1^{i,l}(1:j-1)\right) p\left(y_1(1:j-1) \mid \tilde{x}_1^{i,l}(1:j-1)\right) q_{1,j}\left(x_1^{i,l}(j)\right)}$
8:         end for
9:         Normalize $W_{1,j}^{i,l} \propto G_{1,j}^{i,l}$
10:        Resample $\{W_{1,j}^{i,l},\, x_1^{i,l}(1:j)\}_{l=1}^{M_d}$ to obtain $M_d$ equally weighted particles $\{\frac{1}{M_d}, \check{x}_1^{i,l}(1:j)\}_{l=1}^{M_d}$
11:        Sample $\tilde{x}_1^{i,l}(1:j) \sim K_{1,j}\left(x_1(1:j) \mid \check{x}_1^{i,l}(1:j)\right)$ for $l = 1, ..., M_d$
12:    end for
13:    Calculate $\mathbf{G}_1^i = \prod_{j=1}^d \left(\frac{1}{M_d} \sum_{l=1}^{M_d} G_{1,j}^{i,l}\right)$, the weight associated with the $i^{th}$ particle system
14: end for
15: Calculate $\widehat{p}(y_1) = \frac{1}{N} \sum_{i=1}^N \mathbf{G}_1^i$
16: Normalize $\mathbf{W}_1^i \propto \mathbf{G}_1^i$
17: Resample $\{\mathbf{W}_1^i, \tilde{x}_1^{i,1:M_d}(1:d)\}_{i=1}^N$ to obtain $N$ equally weighted particle systems $\{\frac{1}{N}, \bar{x}_1^{i,1:M_d}(1:d)\}_{i=1}^N$
18:
19: At time step $t \geq 2$:
20: for $i \in \{1, \dots, N\}$ do
21:     for $j \in \{1, \dots, d\}$ do
22:         for $l \in \{1, \dots, M_d\}$ do
23:             Sample $x_t^{i,l}(j) \sim q_{t,j}\left(x_t(j)\right)$
24:             Set $x_t^{i,l}(1:j) \leftarrow \left(\tilde{x}_t^{i,l}(1:j-1),\, x_t^{i,l}(j)\right)$
25:             Calculate $G_{t,j}^{i,l}$ according to (23) if $j = 1$ or (24) if $j \geq 2$
26:         end for
27:         Normalize $W_{t,j}^{i,l} \propto G_{t,j}^{i,l}$
28:         Resample $\{W_{t,j}^{i,l},\, x_t^{i,l}(1:j)\}_{l=1}^{M_d}$ to obtain $M_d$ equally weighted particles $\{\frac{1}{M_d}, \check{x}_t^{i,l}(1:j)\}_{l=1}^{M_d}$
29:         Sample $\tilde{x}_t^{i,l}(1:j) \sim K_{t,j}\left(x_t(1:j) \mid \check{x}_t^{i,l}(1:j)\right)$ for $l = 1, ..., M_d$
30:     end for
31:     Calculate $\mathbf{G}_t^i = \prod_{j=1}^d \left(\frac{1}{M_d} \sum_{l=1}^{M_d} G_{t,j}^{i,l}\right)$, the weight associated with the $i^{th}$ particle system
32: end for
33: Calculate $\widehat{p}(y_{1:t}) = \widehat{p}(y_{1:t-1})\left(\frac{1}{N} \sum_{i=1}^N \mathbf{G}_t^i\right)$
34: Normalize $\mathbf{W}_t^i \propto \mathbf{G}_t^i$
35: Resample $\{\mathbf{W}_t^i, \tilde{x}_t^{i,1:M_d}(1:d)\}_{i=1}^N$ to obtain $N$ equally weighted particle systems $\{\frac{1}{N}, \bar{x}_t^{i,1:M_d}(1:d)\}_{i=1}^N$
```

5. **Static parameter estimation.** In section 3 and 4, we have demonstrated how the standard PF, the STPF and the marginal STPF can be applied to MSV models with cross-leverage effects to estimate the filters and calculate the marginal likelihood. They are all developed on the assumption that the model parameter θ is fixed and known. However, this may not happen in most practical situations since θ is normally unknown. The problem of inferring this unknown θ from the sequence of observations $y_{1:T}$ is known as the problem of static parameter estimation and is of great interests in applied finance [16, 17]. From an Bayesian point of view, SMC methods have been employed together with MCMC algorithms to estimate the static model parameter, which is what we will introduce in this section.

Let $p(\theta)$ be the prior for the unknown parameter θ. Given the sequence of observations $y_{1:T}$, we are interested in estimating or sampling from the posterior distribution $\pi(\theta) = p(\theta|y_{1:T}) \propto p(\theta)p(y_{1:T}|\theta)$. It is straightforward to sample from $\pi(\theta)$ using standard MCMC methods. However, they are idealized MCMC algorithms since the implementation requires the evaluation of the marginal likelihood $p(y_{1:T}|\theta)$ and $p(y_{1:T}|\theta)$ is analytically intractable for MSV models. A popular and direct way to overcome this challenge is to estimate the marginal likelihood unbiasedly by SMC methods and use this estimate within a MCMC framework (e.g. Metropolis-Hastings or the Gibbs sampler). This idea is an exact approximation of the idealized MCMC algorithm and is called the Particle MCMC (PMCMC) method that is proposed in [1]. For a detailed introduction of the PMCMC method, see [1, 12, 16, 25].

If Metropolis-Hastings algorithm is applied to sample from the target $\pi(\theta)$, we will have the following Particle Marginal Metropolis-Hastings (PMMH) algorithm for the inference of θ for MSV models with cross-leverage effects. Here, $q(\theta|\theta')$ denotes the proposal density for θ, conditional on the current parameter estimate θ'.

Algorithm 4 Particle Marginal Metropolis-Hastings (PMMH) algorithm

```
1: At iteration k = 1:
2: Set an arbitraty $\theta(1)$.
3: Run the standard PF or the STPF algorithm with model parameter $\theta(1)$ and
   compute the marginal likelihood estimate $\widehat{p}_{\theta(1)}(y_{1:T})$.
4: At iteration $k \geq 2$:
5: Sample a proposal $\theta' \sim q(\theta|\theta(k-1))$.
6: Run the standard PF or the STPF algorithm with model parameter $\theta'$ and
   compute the marginal likelihood estimate $\widehat{p}_{\theta'}(y_{1:T})$.
7: Set $\theta(k) = \theta'$ and $\widehat{p}_{\theta(k)}(y_{1:T}) = \widehat{p}_{\theta'}(y_{1:T})$ with probability
   $$1 \wedge \frac{\widehat{p}_{\theta'}(y_{1:T})p(\theta')q(\theta(k-1)|\theta')}{\widehat{p}_{\theta(k-1)}(y_{1:T})p(\theta(k-1))q(\theta'|\theta(k-1))},$$
   otherwise set $\theta(k) = \theta(k-1)$ and $\widehat{p}_{\theta(k)}(y_{1:T}) = \widehat{p}_{\theta(k-1)}(y_{1:T})$.
```

Through the PMMH algorithm displayed in Algorithm 4, we are able to obtain a collection of $\theta-$samples from the posterior distribution $\pi(\theta)$. As it is mentioned in [12, 25], the performance of the PMMH algorihtm is closely related to the choice of the proposal density and the variance of the likelihood estimator. In some problems (e.g. the high-dimensional problem we are considering), it may be prohibitively expensive to take the number of particles N large enough to obtain a good performance of the PMMH algorithm.

6. **Numerical results.**

In order to compare the performance of the standard PF, the STPF and the marginal STPF algorithm in estimating the filters and calculating the marginal likelihood for high-dimensional MSV models with cross-leverage effects, both simulation studies and real data analysis are conducted. All algorithms have been implemented in Matlab and the code is available upon request.

6.1. **Simulation studies.**

6.1.1. *Parameter Settings.* Simulation studies are conducted based upon a sequence of observed returns $\{Y_t\}_{t\geq 1}$, which is generated using the MSV model presented in section 2.2 with the following parameter values.

$$\phi_i = 0.91,\ \sigma_{i,\varepsilon\varepsilon} = \sqrt{\text{var}(\varepsilon_{it})} = 1,\ \sigma_{i,\eta\eta} = \sqrt{\text{var}(\eta_{it})} = 1,$$
$$\rho_{i,\varepsilon\eta} = \text{corr}(\varepsilon_{it}, \eta_{it}) = -0.2,\ \text{for } i = 1, ..., d \text{ and } t = 1, ..., T.$$
$$\rho_{ij,\varepsilon\varepsilon} = \text{corr}(\varepsilon_{it}, \varepsilon_{jt}) = 0.2,\ \rho_{ij,\eta\eta} = \text{corr}(\eta_{it}, \eta_{jt}) = 0.3,$$
$$\rho_{ij,\varepsilon\eta} = \text{corr}(\varepsilon_{it}, \eta_{jt}) = -0.1,\ \text{for } i \neq j \text{ and } t = 1, ..., T.$$

The negative values of $\rho_{i,\varepsilon\eta}$ and $\rho_{ij,\varepsilon\eta}$ imply the existence of the leverage and cross-leverage effects respectively.

Table 1. Number of particles used in each algorithm

d	Standard PF	STPF	Marginal STPF
25	$N = 1000$	$N = 50$, $M_d = 20$	$N = 1$, $M_d = 1000$
50	$N = 1000$	$N = 50$, $M_d = 20$	$N = 1$, $M_d = 1000$
100	$N = 1000$	$N = 50$, $M_d = 20$	N.A.
200	$N = 1000$	$N = 50$, $M_d = 20$	N.A.

Table 1 above displays the number of particles that is used in each of the approximation techniques. From this table, we can see that four different values of the dimension of the volatility vector d are considered and these three algorithms are compared based on the same number of particles used. In addition, 300 time steps ($T = 300$) are considered and all the results are summarized over 20 runs of each algorithm. For the marginal STPF, the MCMC resample moves are just Gaussian random walks with the standard deviation (SD) values well tuned so that the acceptance rates lie within 15% to 45%. We have conducted additional experiments with different model parameters. Overall the results seem to be in agreement with the ones presented here.

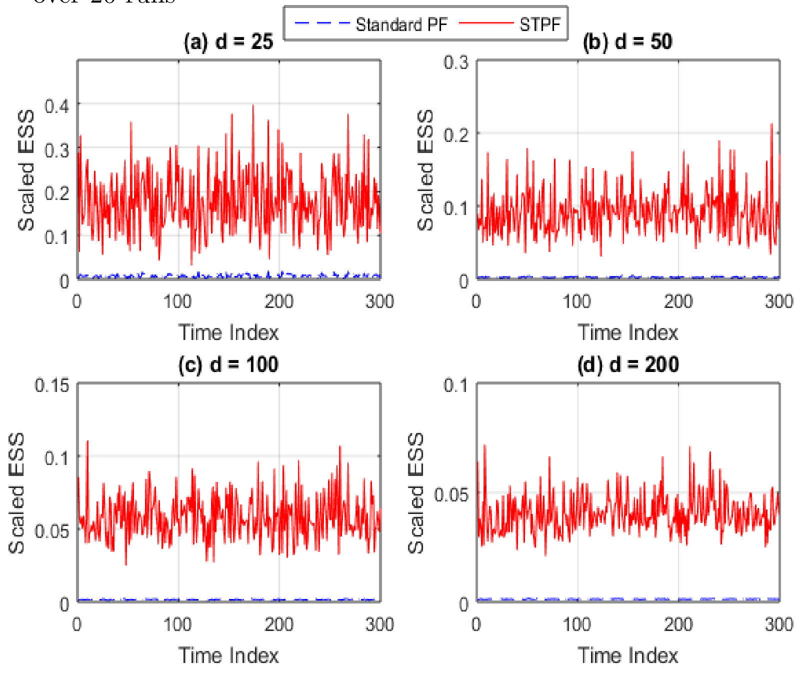
6.1.2. *Simulation Results.* We compare these three algorithms in several aspects:

- The scaled ESS (scaled by the number of particles). It tells us roughly the number of useful samples the algorithm has. An algorithm with approximately 0 as the scaled ESS value means that very few effective samples are being used and thus this algorithm collapses and the results obtained using this algorithm is rather unreliable.
- The bias and stability when estimating the mean of the filters.
- The stability when calculating the marginal likelihood values.

TABLE 2. Computation time (in minutes) per 50 time points for each algorithm

d	Standard PF	STPF	Marginal STPF
25	0.3	2	46
50	0.6	3	110
100	2	16.7	N.A.
200	5	120	N.A.

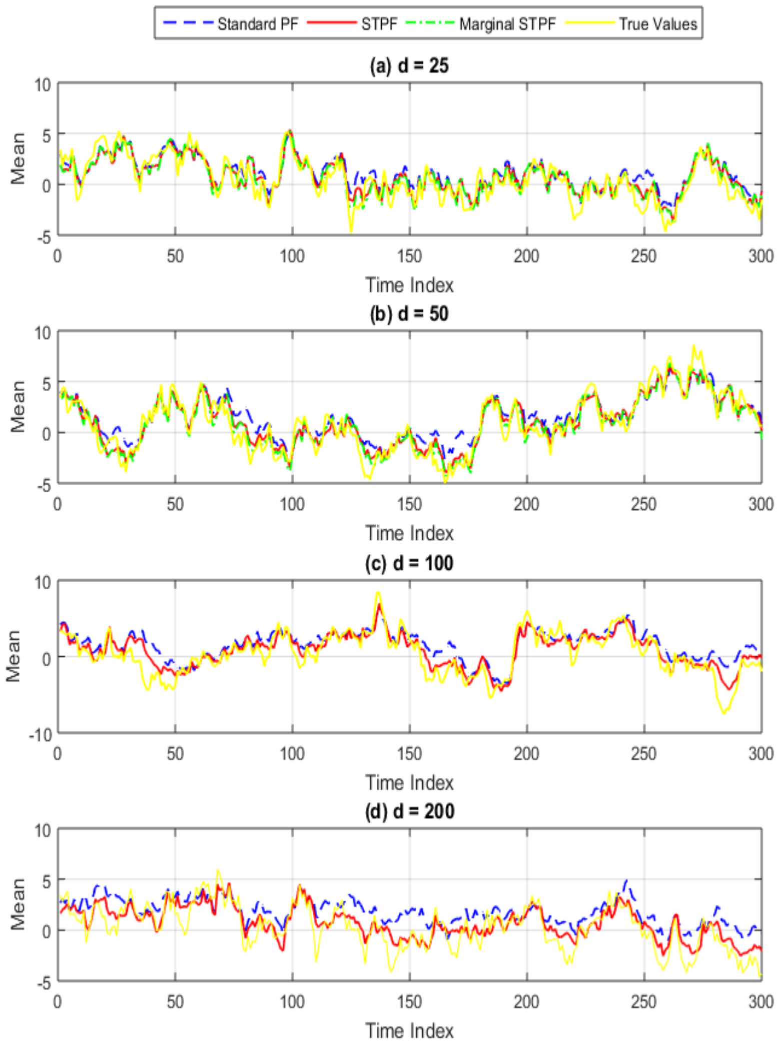
FIGURE 2. Plots of Scaled Effective Sample Size (ESS) averaged over 20 runs



When comparing the standard PF and the STPF across four different values of d, similar results are obtained. First of all, from Figure 2, it is obvious that the standard PF collapses when d increases by presenting extremely low scaled ESS values (i.e. approximately 0). In contrast, the STPF still provides reasonable scaled ESS values. Although one might want these values to be larger, one still can perform estimation in this case, whereas this is not sensible for the PF. Therefore, all the results obtained by the standard PF are not as reliable and credible as those obtained by the STPF. In addition, the STPF outperforms the standard PF in both the accuracy and stability when estimating the filters. This is verified by the smaller bias that the STPF achieves as shown by Figure 3 and the substantial reduction in SD as shown by Figure 4 for the estimation of the mean of the filters. Furthermore, the STPF also improves over the standard PF in terms of marginal likelihood calculation. This is supported by Figure 5, where the STPF achieves much better stability by producing much smaller SD values.

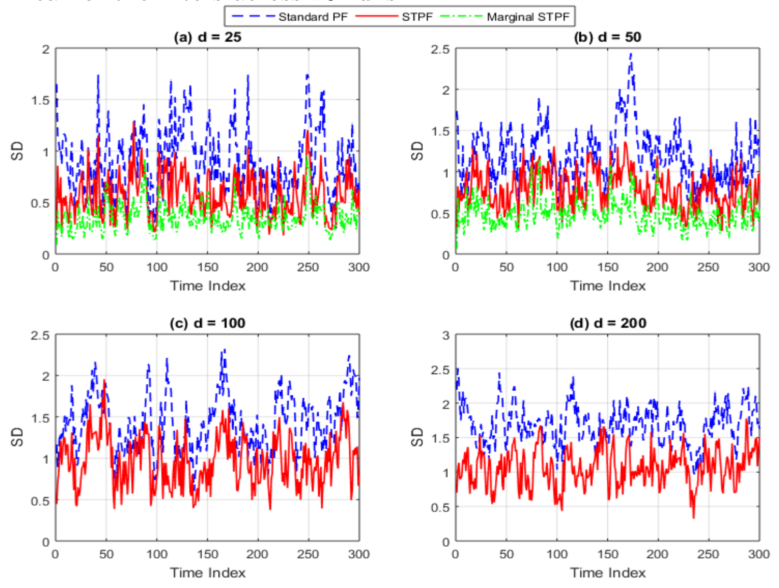
It is worth noting that the results for all the other components of the mean of the filters are similar to the ones presented in Figure 3 and 4, even though only the mean and SD of the estimates for the 1st component are demonstrated for brevity.

FIGURE 3. Plots of mean of estimates for the 1st component of the mean of the filters across 20 runs



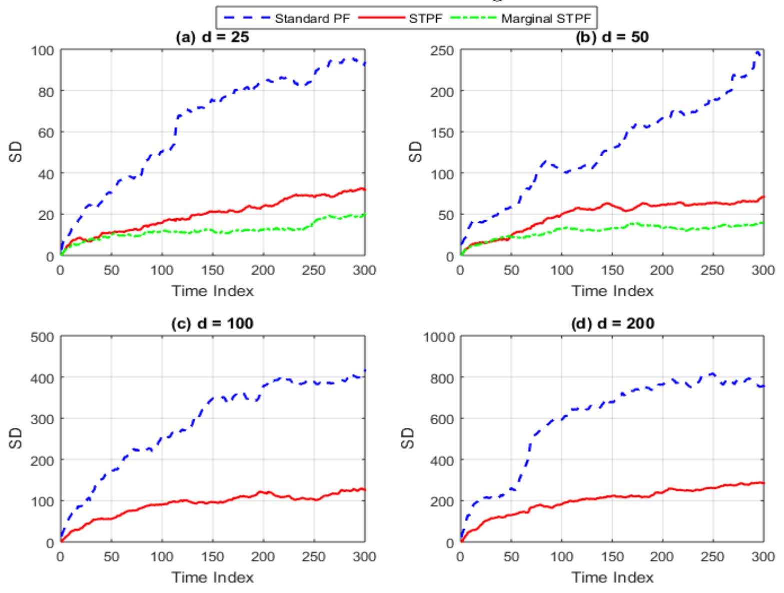
Moreover, when the marginal STPF is implemented for $d = 25$ and $d = 50$ cases, we observe a even greater reduction in SD than the STPF for both the estimation of filters and the calculation of marginal likelihood as illustrated by Figure 4 and 5 respectively. In addition, its estimates for the 1st component of the mean of the filters are the closest to the simulated true values as in Figure 3. Therefore, the marginal STPF algorithm achieves the best performance in these three algorithms

Figure 4. Plots of SD of estimates for the 1st component of the mean of the filters across 20 runs



by providing the most accurate and stable estimations. However, this is at the cost of much longer computational time, which is illustrated in Table 2.

Figure 5. Plots of SD of the estimated log-likelihoods across 20 runs



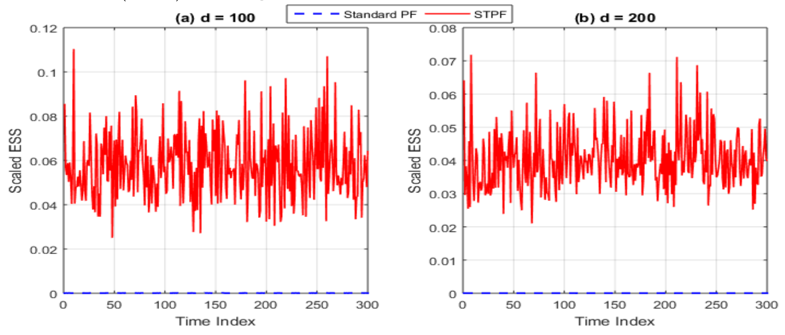
6.1.3. *Time Comparison Study.* Even though better performance is observed for the STPF as compared with the standard PF, the STPF incurs a much higher computational complexity when the same number of particles are used as shown in Table 2. As a result, it would be interesting to compare their performance if they are run for the same amount of computational time. The number of particles and computation time (in minutes) per 50 time points for each algorithm are summarized in Table 3.

TABLE 3. Number of particles and computation time (in minutes) per 50 time points for each algorithm

d	Standard PF	STPF	Computation Time
100	$N = 25000$	$N = 50,\ M_d = 20$	16.7
200	$N = 40000$	$N = 50,\ M_d = 20$	120

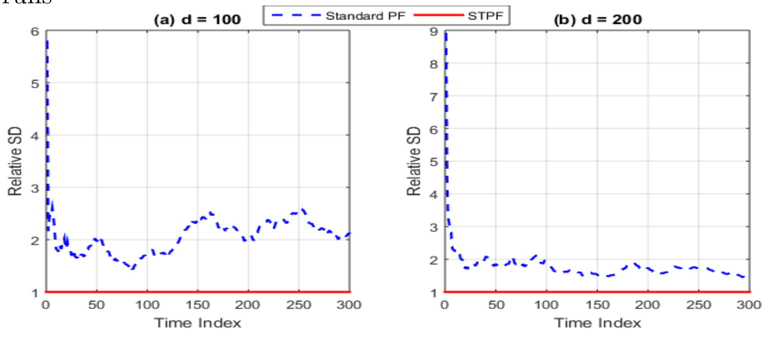
When they are run for the same amount of computational time, exactly the same comparison results hold as in the previous section 6.1.2. Therefore, only selected figures are presented here. First, the collapse and failure of the standard PF is illustrated by the extremely low scaled ESS values in Figure 6. Second, the superiority of the STPF in stability is shown by the smaller SD as in Figure 7.

FIGURE 6. Time Comparison Study Plots of Scaled Effective Sample Size (ESS) averaged over 20 runs



6.1.4. *Parameter Estimation Using PMMH.* We assume that all the 7 parameters in the model set-up in section 6.1.1 are unknown and consider the parameter inference for θ with $\theta = (\phi_i, \sigma_{i,\varepsilon\varepsilon}, \sigma_{i,\eta\eta}, \rho_{ij,\varepsilon\varepsilon}, \rho_{ij,\eta\eta}, \rho_{i,\varepsilon\eta}, \rho_{ij,\varepsilon\eta})'$. We would like to compare the performance of the PMMH algorithm using likelihood estimates from the standard PF and the STPF for the same amount of computational time. For $\phi_i, \rho_{ij,\varepsilon\varepsilon}, \rho_{ij,\eta\eta}, \rho_{i,\varepsilon\eta}$ and $\rho_{ij,\varepsilon\eta}$, a uniform prior on $[-1, 1]$ is used and the proposal is chosen to be a random walk on the logit scale. For $\sigma_{i,\varepsilon\varepsilon}$ and $\sigma_{i,\eta\eta}$, a Gamma prior is used and the proposal is a random walk on the log scale. We run the PMMH algorithm for 8000 iterations with a burn-in period of 500 iterations and initial condition $\theta_0 = (0.85, 1.2, 0.2, 0, 0, 0, 0)'$. Both chains are thinned by collecting every 5^{th} sample. An observation sequence with 50 time points is generated from the model with $d = 10$ and $\theta = (0.91, 1, 0.5, 0.2, 0.3, -0.2, -0.1)'$. For the STPF, we set $N = 100$ and $M_d = 100$ and the number of particles for the standard PF is

FIGURE 7. Time Comparison Study Plots of Relative SD of the estimated log-likelihoods (w.r.t. the SD of the STPF) across 20 runs



adjusted accordingly to allow roughly the same amount of computational time. We display the results for a typical run for both procedures in Figure 8 and 9. Out of the 7 parameters, only the results for $\rho_{ij,\varepsilon\varepsilon}$ and $\sigma_{i,\eta\eta}$ are displayed for brevity. The results for the remaining parameters are similar to the ones presented here.

It is obvious that the results of the PMMH algorithm using likelihood estimates from the STPF are better since the chain mixes well while the chain using likelihood estimates from the standard PF sometimes gets stuck at certain values for quite some time. This is further supported by the autocorrelation (ACF) plots presented in Figure 8 and 9. Given the scaled ESS plot in Figure 10, this is to be expected. As in Figure 10, the standard PF achieves much lower scaled ESS values as compared to the STPF and collapses at some time points because very few effective samples are used. Hence, it provides unstable likelihood estimates with large variance, which results in the poor performance of the PMMH algorithm. This agrees with the discussion in section 5 as well, which states that the performance of the PMMH algorithm is highly related to the variance of the likelihood estimator [12, 25]. Therefore, the STPF outperforms the standard PF by providing more stable estimates for the marginal likelihood and results in much better performance when making inference about the model static parameter.

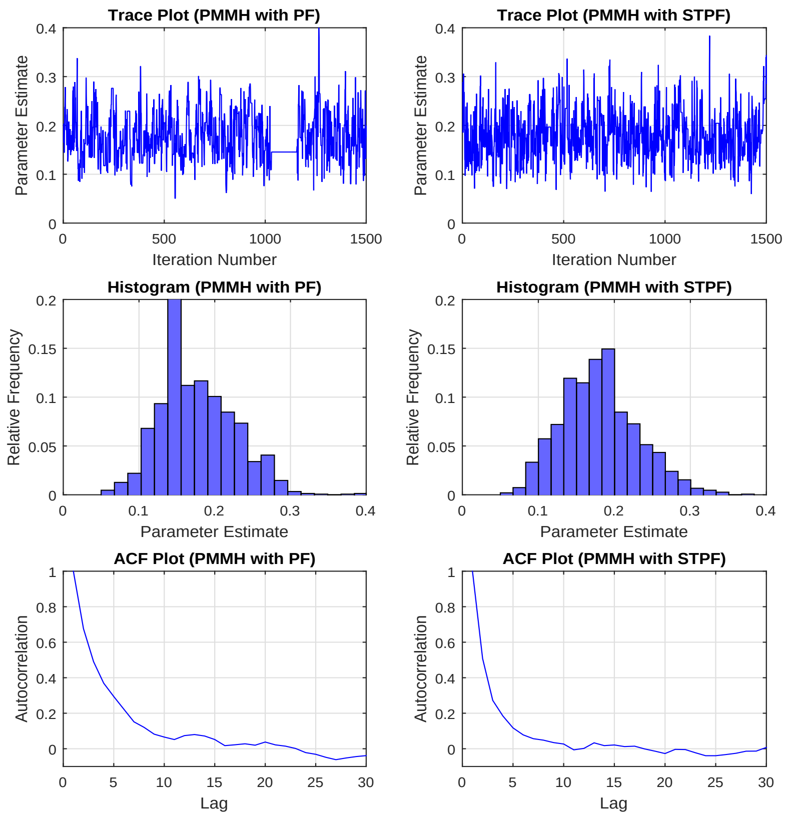
6.2. **Empirical studies.**

6.2.1. *Data Collection and Parameter Settings.* We have collected stock prices data for S&P 500 from January 2, 2012 to December 31, 2015 from Yahoo Finance. The returns are defined by the log difference of the consecutive-day stock prices and then multiplied by 100 for each stock [14]. After data clearance (excluding transaction days with incomplete data entries), it involves 1004 transaction days (i.e. $T = 1004$) in our empirical studies.

In order to compare the performance of these three algorithms, 25, 50, 100 and 200 stocks are chosen at random respectively for analysis. This means that four different values of d are considered. The goal is to estimate the hidden volatility vector at day n based on all the returns data collected up to the n^{th} transaction day and compute the marginal likelihood as well.

The static model parameter values are assumed to be the same as those in section 6.1.1 for convenience and we expect similar results to be obtained if different static model parameter values are used. The number of particles used in each algorithm

FIGURE 8. Trace plots, histograms and ACF plots of the parameter estimates using PMMH for $\rho_{ij,\varepsilon\varepsilon}$



is also displayed in Table 1. For the marginal STPF, Gaussian random walks are conducted as the MCMC resample moves. All the results are summarized over 20 runs of each algorithm.

6.2.2. *Empirical Results.* The ESS plots, the plots of SD of the estimates for the mean of the filters and the plots of SD of the estimated log-likelihoods obtained from the empirical studies have exactly the same patterns as those obtained through simulation in section 6.1.2. As a consequence, the analysis of empirical results stays almost identical to that in section 6.1.2. To summarize, the marginal STPF achieves the best performance by providing the most stable estimates, but its computational time is substantially longer, as displayed in Table 2. The STPF is a good balance between performance and computational cost. Both algorithms improve significantly over the standard PF, suggested by the decent ESS values as shown in Figure 11 and the great reduction in SD as shown in Figure 12.

6.2.3. *Time Comparison Study.* Similar to section 6.1.3, it is interesting and practical to compare the standard PF and the STPF under the assumption that they are

Figure 9. Trace plots, histograms and ACF plots of the parameter estimates using PMMH for $\sigma_{i,\eta\eta}$

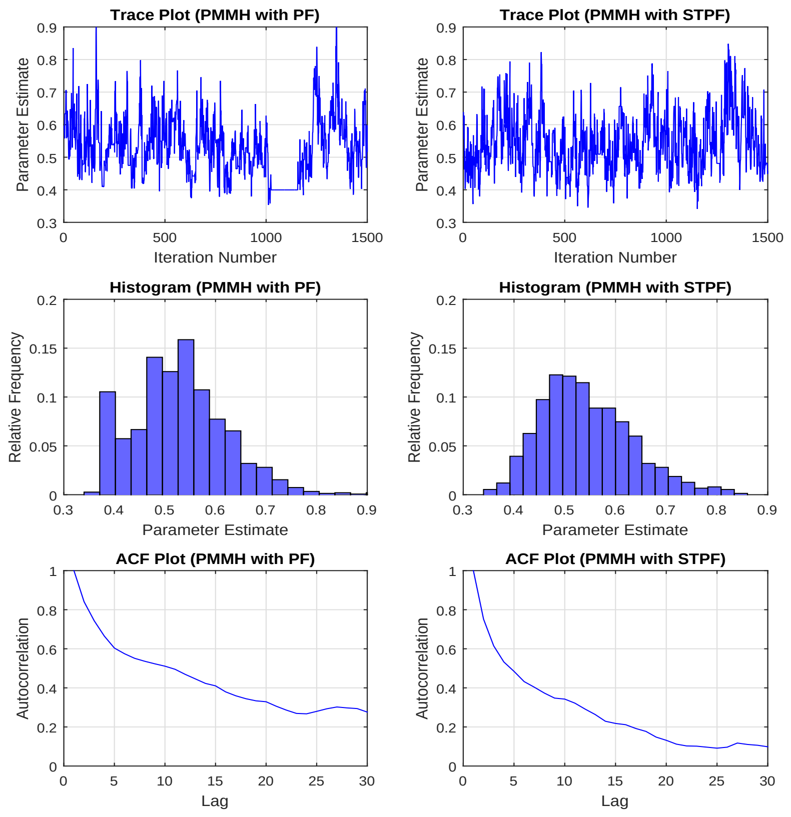
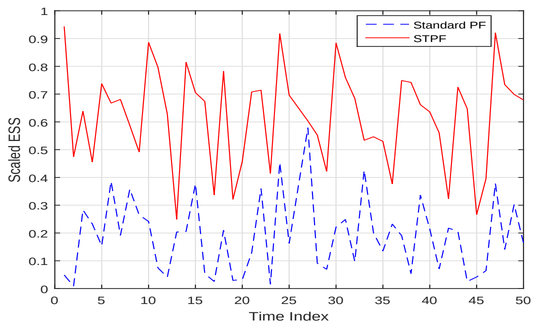


Figure 10. Plot of Scaled Effective Sample Size (ESS)



run for the same amount of time. The number of particles and computation time (in minutes) per 50 time points for each algorithm are demonstrated in Table 3. The same arguments hold when the same amount of computational time is considered

FIGURE 11. Plots of Scaled Effective Sample Size (ESS) averaged over 20 runs

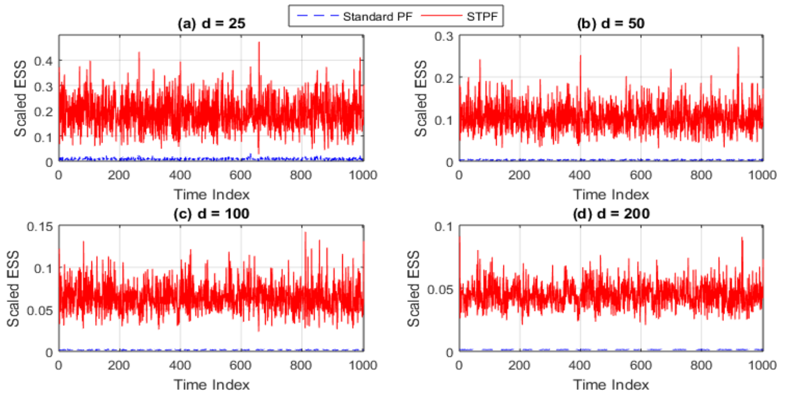
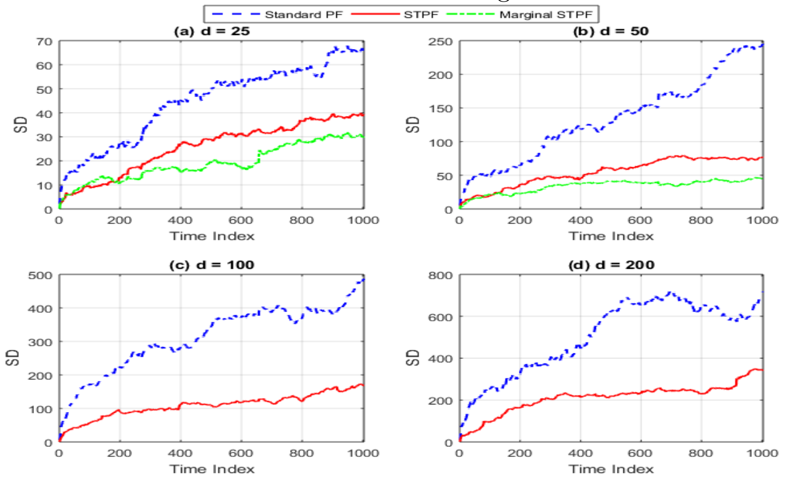


FIGURE 12. Plots of SD of the estimated log-likelihoods across 20 runs



as those in section 6.2.2. Therefore, only Figure 13 and Figure 14 are presented here to illustrate the superiority of the STPF.

7. Conclusions.

We aim to address the challenging problem of high dimensionality when analyzing MSV models. We explore the performance of advances SMC methods, which are the STPF and the marginal STPF in terms of filtering distribution estimation and marginal likelihood calculation, as compared with the standard PF. Our simulation and empirical studies suggest that both the STPF and the marginal STPF improve over the standard PF in the above-mentioned inference problems by producing much more credible, accurate and stable estimators. They work well even in the cases that the standard PF fails completely when a large number of stocks are analyzed simultaneously. The marginal STPF provides the best performance by obtaining the most significant reduction in SD, but it suffers from much heavier computational burden. The STPF is a perfect balance between performance and computational complexity. Moreover, the STPF can be used in combination with

Figure 13. Time Comparison Study Plots of Scaled Effective Sample Size (ESS) averaged over 20 runs

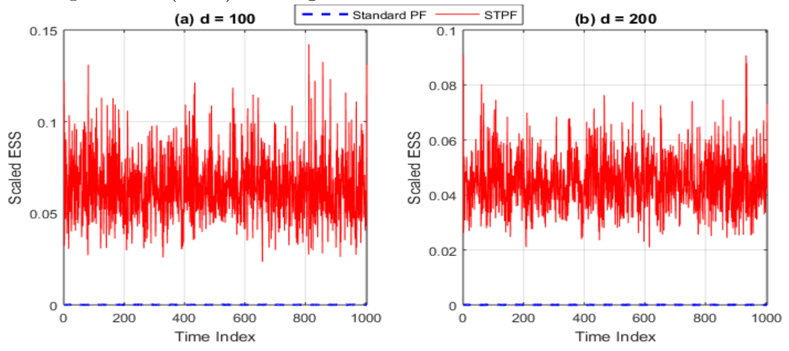
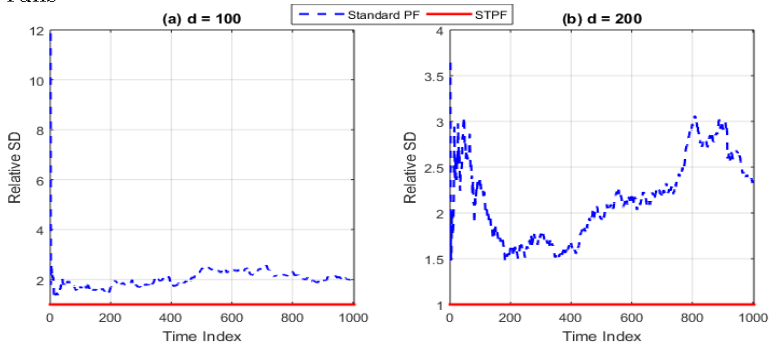


Figure 14. Time Comparison Study Plots of Relative SD of the estimated log-likelihoods (w.r.t. the SD of the STPF) across 20 runs



the PMCMC algorithm to make inference of the static model parameter when the standard PF fails to do so. Therefore, it is definitely a much better choice than the standard PF when investigating high-dimensional MSV models.

As discussed in this article, the standard PF, the STPF and the marginal STPF can be applied to MSV models to capture some important features of the model, e.g. the time-varying volatilities, the leverage and cross-leverage effect and the covariation effect. It is worth noting that all these SMC methods can be further extended to MSV models with heavy-tailed errors to capture the fat-tailed nature of financial returns.

REFERENCES

[1] C. Andrieu, A. Doucet and R. Holenstein, Particle Markov chain Monte Carlo methods (with discussion), *J. R. Statist. Soc. Ser. B*, **72** (2010), 269–342.

[2] M. Asai, M. McAleer and J. Yu, Multivariate stochastic volatility: A review, *Econ. Rev.*, **25** (2006), 145–175

[3] L. Bauwens, S. Laurent and J. V. Rombouts, Multivariate GARCH models: A survey, *J. Appl. Econ.*, **21** (2006), 79–109.

[4] A. Beskos, D. Crisan, A. Jasra, K. Kamatani and Y. Zhou, A stable particle filter for a class of high-dimensional state-space models, *Adv. Appl. Probab.*, **49** (2017), 24–48.

[5] P. Bickel, B. Li and T. Bengtsson, Sharp failure rates for the bootstrap particle filter in high dimensions, In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of J. Ghosh*, IMS, **3** (2008), 318–329.
[6] S. Chib, F. Nadari and N. Shephard, Analysis of high-dimensional multivariate stochastic volatility models, *J. Econ.*, **134** (2006), 341–371.
[7] N. Chopin, Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference, *Ann. Statist.*, **32** (2004), 2385–2411.
[8] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer, New York, 2004.
[9] P. Del Moral, A. Doucet and A. Jasra, On adaptive resampling strategies for sequential Monte Carlo methods, *Bernoulli*, **18** (2012), 252–278.
[10] A. Doucet, *On Sequential Simulation-based Methods for Bayesian Filtering*, Technical Report, 1998.
[11] A. Doucet and A. Johansen, A tutorial on particle filtering and smoothing: Fifteen years later, In *Handbook of Nonlinear Filtering* (eds. D. Crisan & B. Rozovsky), Oxford University Press, Oxford, (2011), 656–704.
[12] A. Doucet, M. K. Pitt, G. Deligiannidis and R. Kohn, Efficient Implementation of Markov chain Monte Carlo when Using an Unbiased Likelihood Estimator, *Biometrika*, **102** (2015), 295–313.
[13] J. Hull and A. White, The pricing of options on assets with stochastic volatilities, *J. Finan.*, **42** (1987), 281–300.
[14] T. Ishihara and Y. Omori, Efficient Bayesian estimation of a multivariate stochastic volatility with cross leverage and heavy tailed errors, *Comp. Statist. Data Anal.*, **56** (2012), 3674–3689.
[15] A. Jasra, D. A. Stephens, A. Doucet and T. Tsagaris, Inference for Lévy driven stochastic volatility models via adaptive sequential Monte Carlo, *Scand. J. Statist.*, **38** (2011), 1–22.
[16] N. Kantas, A. Doucet, S. S. Singh, J. M. Maciejowski and N. Chopin, An overview of sequential Monte Carlo methods for parameter estimation in general state-space sodels, *IFAC Proc.*, **42** (2009), 774–785.
[17] N. Kantas, A. Doucet, S. S. Singh, J. M. Maciejowski and N. Chopin, On particle methods for parameter estimation in general state-space models, *Statist. Sci.*, **30** (2015), 328–351.
[18] S. Kim, N. Shephard and S. Chib, Stochastic volatility: Likelihood inference and comparison with ARCH models, *Rev. Econ. Stud.*, **65** (1998), 361–393.
[19] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state-space models, *J. Comp. Graph. Stat.*, **5** (1996), 1–25.
[20] M. Klaas, N. De Freitas and A. Doucet, Towards practical N^2 Monte Carlo: The marginal particle filter, *Uncert. A. I.*, (2005), 308–315.
[21] A. Kong, J. S. Liu and W. H. Wong, Sequential imputations and Bayesian missing data problems, *J. Amer. Statist. Assoc.*, **89** (1994), 278–288.
[22] C. Naesseth, F. Lindten and T. Schön, Nested sequential Monte Carlo methods, *ICML*, (2015), 1292–1301.
[23] J. Nakajima, Bayesian analysis of multivariate stochastic volatility with skew return distribution, *Econ. Rev.*, **36** (2017), 546–562.
[24] S. S. Ozturk and J. F. Richard, Stochastic volatility and leverage: Application to a panel of S & P 500 stocks, *Finan. Res. Lett.*, **12** (2015), 67–76.
[25] M. K. Pitt, R. Dos Santos Silva, P. Giordani and R. Kohn, On some properties of Markov chain Monte Carlo simulation methods based upon the particle filter, *J. Econom.*, **171** (2012), 134–151.
[26] M. K. Pitt and N. Shephard, Filtering via simulation: Auxiliary particle filters, *J. Amer. Statist. Assoc.*, **94** (1999), 590–599.
[27] K. Platanioti, E. McCoy and D. A. Stephens, *A Review of Stochastic Volatility Models*, Technical Report, 2005.
[28] C. Snyder, T. Bengtsson, P. Bickel and J. Anderson, Obstacles to high-dimensional particle filtering, *Month. Weather Rev.*, **136** (2008), 4629–4640.
[29] C. Vergé, C. Duberry, P. Del Moral and E. Moulines, On parallel implementation of sequential Monte Carlo methods: The island particle filtering, *Stat. Comp.*, **25** (2015), 243–260.

E-mail address: a0078115@u.nus.edu
E-mail address: staja@nus.edu.sg