Stochastic Volatility Inference with Monte Carlo Filters

Nikolay Y. Nikolaev
Department of Computing
Goldsmiths College, University of London
London SE14 6NW
United Kingdom
N.Nikolaev@gold.ac.uk

Evgueni Smirnov
Faculty of General Sciences
Maastricht University
Maastricht
The Netherlands
Smirnov@cs.unimaas.nl

Abstract

Modelling the volatilities in business cycles is a problem whose solutions have important implications for asset pricing, risk management, and other financial tasks. There are various approaches to finding the volatility, considered as an an unobsrved component in financial time series, but the practitioners usually take only the most popular ones. This paper shows that even popular Monte Carlo filters can be outperformed by modern tools product of recent research. A mixture particle filter is presented and applied to infer volatilities from stock returns, adopting a general stochastic volatility model.

1 Introduction

Accurate pricing of financial instruments and sucessfull trading can be achieved using estimates of the volatility. Volatility is called the uncertainty, that is the deviation of particular stock returns in the market. Decsriptions of returns are provided by stochastic volatility models [Ghysels et al., 1996], [Poon and Granger, 2003], [Shephard, 2005], in which the volatility is an unknown parameter and can be inferred [Harvey et al., 1994], [Pitt and Shephard, 1999], [Durbin and Koopman, 2000], [Jacquier et al., 2002]. The stochastic volatility models are found empirically more successfull than standard GARCH models [Kim et al., 1998], and they capture better second-order properties of the returns as well as leverage effects.

Finding the stochastic volatility from series of stock prices involves learning of nonlinear models. A popular approach for nonlinear volatility modelling are particle filters (PF) [Gordon et al., 1993], [Kitagawa, 1996], [Liu and Chen, 1998], [Carpenter et al., 1999], [Bauwens et al., 1999], [Doucet et al., 2001], [Maskell, 2004], [Stroud et al., 2004]. The particle filter is a Bayesian mechanism that uses a population of Monte Carlo samples, called particles, whose averaging allows to compute the characteristics (mean and deviation) of the posterior distribution of the unknown parameters. Having such characteristics one may generate forecasts from unseen inputs and evaluate confidence intervals. The particle filters are suitable for stochastic volatility modelling because they can work reliably in relaxed circumstances, like non-Gaussianity and non-stationarity.

A serious practical problem of particle filters when applied to financial time series is that the typically present outliers disturb their performance, and prevent them from producing good results. The main reason is that they are relatively sensitive to the magnitude of outliers. There are various extensions to make these tools more robust, the most popular of which is the Auxiliary Particle Filter (APF) [Pitt and Shephard, 1999]. APF boosts particles with higher predictive likelihood to reduce the variance of the hidden parameters in order to attain a better description of the posterior. However if there is a large amount of process noise and outliers, the auxiliary filter is not much better than a generical particle filter, and this limits its usefulness.

A promising contemporary approach to sequential Bayesian inference, that has not been investigated yet on finacial time series, are the mixture Monte Carlo particle filters. Recently efficient Gaussian Particle Filters (GPF) [Kotecha and Djuric, 2003a] from this class were proposed for signal processing. The GPF employ particle filtering to compute the mean and variance of the relevant distributions, and propagate only these two moments recursively in time. Although GPF approximates the distributions by Gaussians they can be made to model satisfactory many kinds of data. Accurate tracking of even non-Gaussian processes can be performed by extending the GPF to use mixtures of Gaussian distributions.

Here we demonstrate a Gaussian Sum Particle Filter (GSPF) [Kotecha and Djuric, 2003b] which uses Gaussian mixtures to represent all the distributions of interest, including the noise on the latent parameters. Representing even the noise on the unobserved parameters by a mixture density increases the descriptive potential of the filter, and makes it more flexible when processing oscillating series. This investigation shows that the GSPF filter is suitable for modelling financial data.

The task of learning a general stochastic volatility model from a given series of stock returns is considered. There are implemented a particle filter PF, an auxiliary particle filter APF, and a Gaussian sum particle filter GSPF. These filters are examined on two series: an artificially generated time series of log returns, and a real-world series of exchange rates.

This article outlines the selected volatility model in section two. Section three offers the basics of sequential particle filtering and the auxiliary particle filtering approach. After that, the mixture particle filters are developed. The next section five provides the experimental results. Finally a brief conclusion is made.

2 The Stochastic Volatility Model

The option price theory [Hull, 2005] assumes that stock prices follow a geometric Brownian motion. In general, the stock price depends on the current price, the risk-free interest rate, the volatility, the strike price and the time to maturity. Among these factors only the volatility (variance of the stock return) can not be observed, and often treated as a constant. In reality, however, the volatility is also a stochastic variable and could be inferred adopting some pricing model. The choice of the model is of crucial importance for finding the volatility, because the format of the model determines which could be the appropriate learning algorithm and, hence, impacts the success of the learning process.

A stochastic volatility model consists of a couple of equations that describe how the returns depend on the volatility. We consider a general model which expresses the log returns $y_t \equiv \log(S_t) - \log(S_{t-1})$ on stock prices S_t and S_{t-1}, as a nonlinear function of the log of the volatility $x_t \equiv \log(v)$ at time t with the following equations [Hull and White, 1987], [Shephard, 1993], [Kim et al., 1998]:

$$x_t = \gamma + \phi(x_{t-1} - \gamma) + \varepsilon_t \quad (1)$$

$$y_t = \eta_t \exp(x_t/2) + \mu \quad (2)$$

where γ is a level constant, ϕ is the persistence of the volatility, ε_t is a univariate Gaussian noise $\varepsilon_t = \mathcal{N}(0, q)$, and η_t is a scaling factor for the noise of the log-returns, which is also proportional to Gaussian $\eta_t = \mathcal{N}(0, 1)$.

This model has several characteristics: 1) the volatility x_t is a linear autoregressive function $x_t = h(x_{t-1}) + \varepsilon_t$ (1) (state equation); 2) the state noise ε_t is zero-mean Gaussian but with an unknown variance q; 3) the output is a nonlinear function $y_t = g(x_t) + \mu$ of the latent volatility x_t (2) (observation equation); 4) the distribution of the returns y_t has a constant mean μ, i.e. $y_t = \mathcal{N}(\mu, v)$, while its variance x_t is changing around some predefined level γ at rate ϕ; and 5) the returns y_t are considered independent, and normally distributed.

Equations (1) and (2) provide a general state-space model formulation which should be learnable using Kalman filters, however there is an intrinsic problem that prevents this. The problem is that the returns and the volatility become uncorrelated when treated with the linear Kalman equations. Not only Kalman filters [Harvey, 1989], but even derivative-free filters [Norgaard et al., 2000], [Wan and de Merwe, 2001] fail to estimate the volatility with the above model. Although such sophisticated filters have been applied to some specific models [Javaheri et al., 2003], little research has been done to study other more universal filters that can deal with this general log volatility model. Novel sequential Bayesian filters that have the potential to deal with the general log volatility model are investigated here.

3 Sequential Bayesian Estimation

The Bayesian framework helps to make filters that infer averaged statistics, in the sense of finding the mean and variance of the state posterior distribution from the provided data. The probability distributions decsribe uncertainties, and inference is performed by applying probabilistic rules to them. The the Bayes' rule (theorem) shows how to update the belief in the state after seeing the data through their likelihood. The initial belief in the state is encoded into a prior.

A sequential probabilistic filter applies the Bayes' theorem recursively in time to obtain the value of the state variable x_t as a sufficient statistic of the whole history $\mathbf{x}_{0:t}$ up to time t:

$$p(x_t|\mathbf{y}_{0:t}) = \frac{p(y_t|x_t)p(x_t|\mathbf{y}_{0:t-1})}{p(y_t|\mathbf{y}_{0:t-1})} \tag{3}$$

where $p(x_t|\mathbf{y}_{0:t})$ is the state posterior (filtering distribution), $p(y_t|x_t)$ is the likelihood of the data that reflects the model discrepancy, $p(x_t|\mathbf{y}_{0:t-1})$ is the state prior (predictive distribution), and $p(y_t|\mathbf{y}_{0:t-1})$ is the evidence for the model. The state prior is defined by the integral: $p(x_t|\mathbf{y}_{0:t-1}) = \int p(x_t|x_{t-1}, \mathbf{y}_{0:t-1})p(x_{t-1}|\mathbf{y}_{0:t-1})dx_{t-1}$, where $p(x_{t-1}|\mathbf{y}_{0:t-1})$ is the previous filtering density, and $p(x_t|x_{t-1}, \mathbf{y}_{0:t-1})$ is the transition density.

It is difficult, however, to make such a filter for the general log volatility model (1) and (2) so as to learn the state representing the volatility. The reason is that equation (2) is a non-linear model, leading to nonstandard distributions (without a predominant mode), and for such cases a close analytical solution for the state posterior can not be derived. This difficulty to obtain the posterior can be alleviated using two approaches: 1) using parametric Gaussian approximation; or 2) using Monte Carlo sampling (non-parametric approximation).

The sophisticated filters that exploit the Gaussian approximation, like the Unscented Kalman filter (UKF) [Wan and de Merwe, 2001] and the Central Difference Filter (CDF) [Norgaard et al., 2000], are computationally efficient. They achieve higher accuracy than extended Kalman filters because they calculate the predictive state density by deterministic sampling, but after that they estimate the posterior with the standard linear equations. It has been proven theoretically that when applied to the above stochastic log volatility model the linearization of the output model leads to uncorrelated state and the outputs, and the state is not updated [Zoeter et al., 2005]. Therefore, these filters can not be used to learn such stochastic models.

The Monte Carlo methods [Doucet et al., 2001] compute reliably both the predictive and filtering distributions by averaging over populations of samples. These methods are less sensitive to the specificities in the provided training data and fit well the distributions. This makes them especially suitable for processing financial time series, which are typically contaminated by non-Gaussian noise.

4 Monte Carlo Filtering

Sequential Bayesian learning of the probability distributions of interest through Monte Carlo simulation can be performed using particle filters (PF) [Gordon et al., 1993], [Kitagawa, 1996], [Liu and Chen, 1998], [Carpenter et al., 1999], [Bauwens et al., 1999], [Doucet et al., 2001], [Maskell, 2004]. Monte Carlo is a method for calculating the expectation of a probability density function by sampling. According to it the probability density function is represented by a population of random samples (particles) each with a corresponding weight, and its representation is obtained by summation over the weighted samples. This approach allows to approximate arbitrary well any general distribution by increasing the number of samples.

4.1 Monte Carlo Approximation

The Monte Carlo method approximates the state posterior by a function $f(x_t)$ on a discrete support with finite number i of samples $x_t^{(i)}$. Being interested in finding the the mean $\langle f(x_t) \rangle$ of the state posterior $p(x_t|\mathbf{y}_{0:t})$ at time t, it is necessary to generate state samples $x_t^{(i)} \sim p(x_t|\mathbf{y}_{0:t})$ and to calculate their averaged statistic. In practice it is often extremely hard to sample directly from the state distribution. That is why, it is replaced by another proposal (importance) state distribution π, proportional the the true distribution at each point

$\pi(x_t|\mathbf{y}_{0:t}) \propto p(x_t|\mathbf{y}_{0:t})$, to facilitate easy sampling. This leads to the following empirical mean estimate:

$$\begin{aligned} \langle f(x_t) \rangle &= \int f(x_t) \frac{p(x_t|\mathbf{y}_{0:t})}{\pi(x_t|\mathbf{y}_{0:t})} \pi(x_t|\mathbf{y}_{0:t}) dx_t \\ &\approx \frac{\sum_{i=1}^{N} f(x_t^{(i)}) w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}} \end{aligned} \tag{4}$$

where $w_t^{(i)}$ are called importance weights, and N is the number of considered particles.

It is proven that this average converges asymptotically to the expectation of the posterior when increasing the number of samples according to the central limit theorem [Geweke, 1989].

4.2 Sequential Particle Filtering

The sequential particle filtering process involves two stages: 1) generation of the prior state distribution (prediction), and 2) estimation of the state posterior distribution (updating). The operation of the filter begins with random generation of an initial state from its prior. The first prediction stage draws sample states from the proposal distribution $x_t^{(i)} \sim \pi(x_t|\mathbf{y}_{0:t})$. The second updating stage passes these particles $x_t^{(i)}$ to compute outputs $\widehat{y}_t^{(i)}$ using the observation equation (2), evaluates the model likelihoods $p(y_t|x_t^{(i)})$, and updates with these likelihoods the weights from $w_{t-1}^{(i)}$ to $w_t^{(i)}$.

The weight adaptation formula is derived from the definition $w_t = p(x_t|\mathbf{y}_{0:t})/\pi(x_t|\mathbf{y}_{0:t})$. Since the intention is to perform recursive learning two assumptions are made [Doucet et al., 2001]: the proposal distribution can be factored $\pi(x_t|\mathbf{y}_{0:t}) = \pi(x_t|x_{t-1}, \mathbf{y}_{0:t})\pi(x_{t-1}|\mathbf{y}_{0:t-1})$, so as to express the current state dependant on the last state, and the current state depends only on the most recent observation $\pi(x_t|x_{t-1}, \mathbf{y}_{0:t}) = \pi(x_t|x_{t-1}, y_t)$. This leads to the following weight modification equation:

$$w_t^{(i)} \approx w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|x_{t-1}^{(i)}, y_t)} \tag{5}$$

where $p(y_t|x_t^{(i)}) \approx \exp(-0.5 \sum_{t=1}^{T} (y_t - \widehat{y}_t^{(i)})^2)$ is the likelihood and T is the series size. Thus, a particular weight reflects the density in the state space in the vicinity of its corresponding particle.

The empirical estimate of the state posterior is computed by discrete summation as follows:

$$\widehat{p}(x_t|\mathbf{y}_{0:t}) \simeq \sum_{i=1}^{N} W_t^{(i)} \delta(x_t - x_t^{(i)}) \tag{6}$$

where δ is the Dirac delta function, $W_t^{(i)}$ are the normalized importance ratios $W_t^{(i)} = w_t^{(i)} / \sum_{i=1}^{N} w_t^{(i)}$.

The approximation quality of this sequential sampling approach is sensitive to the form of the proposal density. A popular choice is to select as proposal density the transition probabilty of the states $\pi(x_t|x_{t-1}, y_t) =$

$p(x_t|x_{t-1})$ [Kitagawa, 1996], [de Freitas et al., 2000]. This choice has two advantages: it minimizes the variance of the weights, which is necessary to achieve asympthotic convergence of the learning process, and it is easy to implement by making each sample by passing the previous state through the state equation and adding noise ε_t. Then the weight adaptation remains simply $w_t^{(i)} \approx w_{t-1}^{(i)} p(y_t|x_t^{(i)})$.

The problem of this approach is that it tends to degenerate, because the population becomes dominated by many particles with negligible weights. That is why, the states are resampled to retain the most promising among them [Gordon et al., 1993]. Importance resampling is performed by selecting particles with high weights and discarding some of those with insignificant weights. This efficient algorithm is known as Monte Carlo with Sampling Importance Resampling (SIR). The resampling can be implemented with different techniques: residual [Liu and Chen, 1998], multinomial [Carpenter et al., 1999], and systematic [Kitagawa, 1996].

4.3 The Auxiliary Particle Filter

The generic particle filter is weak in handling data with heavy tail distributions. An approach to reducing the effects of outliers is provided by the popular Auxiliary Particle Filter (APF) [Pitt and Shephard, 1999]. Its idea is to push the particles into state regions with high probability density. This is achieved by doing prelimary resampling to simulate good particles using auxiliary indices. Resampling is performed with respect to point estimates associated with the transition densities of the particles.

The filter is derived after rewriting the filtering density $p(x_t|\mathbf{y}_{0:t}) = p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|\mathbf{y}_{0:t-1}) dx_{t-1}$ in a discrete format $p(x_t|\mathbf{y}_{0:t}) \simeq \sum_{j=1}^{N} w_{t-1}^{(j)} p(y_t|x_t^{(j)}, i^j) p(x_t|x_{t-1}^{(j)})$. The density is expressed as dependent on auxiliary variables i that are indices of the particles from the previous time step $t-1$. The discrete modelling of the density $p(x_t|\mathbf{y}_{0:t})$ is realized with the following joint proposal distribution:

$$q(x_t, i|\mathbf{y}_{0:t}) \simeq q(i|\mathbf{y}_{0:t}) q(x_t|i, \mathbf{y}_{0:t}) \tag{7}$$

where the indices i can be thought as made with probability $q(i|\mathbf{y}_{0:t}) \simeq w_{t-1}^{(i)} p(\mathbf{y}_{0:t}|\mu_t^{(i)})$, and the predicted states are inferred using them with probability $q(x_t|i, \mathbf{y}_{0:t}) = p(x_t|x_{t-1}^{(i)})$. Here $\mu_t^{(i)}$ is a point estimate which could be a mean $\mu_t^{(i)} = E[x_t|x_{t-1}^{(i)}]$ or a sample value $\mu_t^{(i)} \sim p(x_t|x_{t-1}^{(i)})$.

Adopting this factorized proposal suggests to obtain initially point estimates $\mu_t^{(i)}$ that characterise the transition prior $p(x_t|x_{t-1}^{(i)})$, in order to evaluate the predictive likelihoods of the particles $p(y_t|\mu_t^{(i)})$. They are taken to compute simulation weights $w_{t-1}^{(i)} p(y_t|\mu_t^{(i)})$ which, after normalization, are passed to a sampling algorithm for drawing states $x_t^{(j)} \sim q(x_t|i^j, \mathbf{y}_{0:t}) = p(x_t|x_{t-1}^{(i^j)})$. These simulated particles $\{x_t^{(j)}, i^j\}_{j=1}^{N}$, with indices i^j from the last time step, are likely to be closer to the true state.

The simulated particles are taken to approximate the state posterior distribution as follows:

$$\widehat{p}(x_t|\mathbf{y}_{0:t}) \simeq \sum_{j=1}^{N} w_{t-1}^{(j)} p(y_t|\mu_t^{(j)}) p(x_t|x_{t-1}^{(j)}) \tag{8}$$

and, next, the importance weights are updated recursively with the formula:

$$\begin{aligned} w_t^{(j)} &\approx w_{t-1}^{(j)} \frac{p(y_t|x_t^{(j)}) p(x_t^{(j)}|x_{t-1}^{(i^j)})}{q(x_t^{(j)}, i^j|\mathbf{y}_{0:t})} & (9) \\ &= \frac{p(y_t|x_t^{(j)})}{p(y_t|\mu_t^{(i^j)})} & (10) \end{aligned}$$

using the point estimates $\mu_t^{(i^j)}$ of the chosen particles.

APF is essentially a one-step ahead procedure that often produces lower variance results than generic PF. However, in situations when the process noise is large, the preliminary resampling may fail to focus on more likely particles. Then, the predictive likelihoods are less sensitive to the states and the APF performance is not much superior than this of the particle filter. A computational drawback of the APF algorithm is that it actually samples twice as it computes twice likelihoods and weights, and its speed is slow.

5 Mixture Particle Filtering

The Gaussian particle filters (GPF) [Kotecha and Djuric, 2003a, 2003b] are tools that perform Monte Carlo approximation of both the predictive and filtering distributions, and sample from a simulated normal state prior. The distributions of interest are represented by mixtures of Gaussians, using non-recursively calculated weights to avoid degeneration of the learning process. The mixtures are computed by particle filtering techniques, while only their mean and variances are propageted sequentially in time. Adopting mixture distributions helps to produce more accurate results because they provide potential to model successively even non-Gaussian densities. Considered for the studied log volatility model, using mixtures in both the prediction and update stages allows to achieve reliable state inference, as problems from uncorrelated state and outputs are avoided.

5.1 Gaussian Particle Filtering

The predictive density can be expressed as a Gaussian mixture from finite number of terms [Kotecha and Djuric, 2003a]. This idea is based on theorems from classical filtering theory which state that the prediction and filtering densities in dynamical models, like these defined by equations (1) and (2), can be described arbitrarily closely by a sufficient number of mixtures from normal distributions [Anderson and Moore, 1979].

The GPF approximates the predictive distribution $p(x_t|\mathbf{y}_{0:t-1}) = \int p(x_t|x_{t-1})\mathcal{N}\left(x_{t-1}; \widehat{x}_{t-1}, P_{t-1}\right) dx_{t-1}$, as a weighted mixture of states produced using samples from the previous time step:

$$\widehat{p}(x_t|\mathbf{y}_{0:t-1}) \simeq \sum_{m=1}^{M} w_{t-1}^{(m)} p\left(x_t|x_{t-1}^{(m)}\right) \tag{11}$$

where the subscript m enumerates the mixture components $1 \leq m \leq M$.

The implementation of this filtering stage begins by drawing state samples from the corresponding previous density $x_{t-1|t-1}^{(m)} \sim \mathcal{N}\left(x_{t-1}; \widehat{x}_{t-1|t-1}^{(m)}, P_{t-1|t-1}^{(m)}\right)$, where $\widehat{x}_{t-1|t-1}^{(m)}$ denotes the mean and $P_{t-1|t-1}^{(m)}$ denotes the covariance matrix. Next, the samples are passed through the state function $x_{t|t-1}^{(m)} = h(x_{t-1|t-1}^{(m)})$ in order to compute the prior $\widehat{p}(x_t|\mathbf{y}_{0:t-1})$. The mean state prediction is obtained by summation: $\widehat{x}_{t|t-1} = \sum_{m=1}^{M} w_{t-1}^{(m)} x_{t|t-1}^{(m)}$, and the covariance matrix is made by differencing: $P_{t|t-1} = \sum_{m=1}^{M} w_{t-1}^{(m)} (x_{t|t-1}^{(m)} - \widehat{x}_{t|t-1})(x_{t|t-1}^{(m)} - \widehat{x}_{t|t-1})^T$.

The second updating stage begins with random sampling of states $x_{t|t}^{(m)}$ from the mixture distribution $\widehat{p}(x_t|\mathbf{y}_{0:t-1}) \sim \mathcal{N}\left(x_{t|t}^{(m)}; \widehat{x}_{t|t-1}, P_{t|t-1}\right)$. These samples are taken to adapt the weights non-recursively, and to modify the state mean and covarianceto in order to absorb the information from the current observation y_t. Even when working with a linear state equation this mixturing strategy is useful as it makes the filter more flexible and tuneable to different kinds of noise. For example, in the studied here general log volatility model the state equation is linear with Gaussian noise, which however has an unknown variance.

Having the characteristics of the predictive distribution $\widehat{p}(x_t|\mathbf{y}_{0:t-1})$, the weights are updated with the following formula:

$$w_t^{(m)} \approx \frac{p(y_t|x_{t|t}^{(m)})\mathcal{N}(x_{t|t}^{(m)}; \widehat{x}_{t|t-1}, P_{t|t-1})}{\pi(x_{t|t}^{(m)}; x_t|\mathbf{y}_{0:t-1})} \tag{12}$$

which is proportional only to the likelihood $w_t^{(m)} \approx p(y_t|x_{t|t}^{(m)})$ when the transition probabilty is used as a proposal. The likelihood is estimated by passing the state samples $x_{t|t}^{(m)}$ through the observation equation (2).

Finally, the filtering density is inferred as a discrete analog of the exact posterior with another Gaussian mixture, using the same state samples $x_{t|t}^{(m)}$ drawn from the predictive mixture distribution:

$$\widehat{p}(x_t|\mathbf{y}_{0:t}) \simeq \sum_{m=1}^{M} w_t^{(m)} \mathcal{N}\left(x_{t|t}^{(m)}; \widehat{x}_{t|t-1}, P_{t|t-1}\right) \tag{13}$$

where the updated mean and covariance are computed by summation:

$$\widehat{x}_{t|t} = c \sum_{m=1}^{M} w_t^{(m)} x_{t|t}^{(m)} \tag{14}$$

$$P_{t|t} = c \sum_{m=1}^{M} w_t^{(m)} (x_{t|t}^{(m)} - \widehat{x}_{t|t})(x_{t|t}^{(m)} - \widehat{x}_{t|t})^T \tag{15}$$

where c is a normalizing constant defined by the inverse $c = [\sum_{m=1}^{M} w_t^{(m)}]^{-1}$.

It has been shown [Kotecha and Djuric, 2003a] that the posterior mean computed in this way converges asympthotically to the minimum mean squared error estimate when a sufficient number of mixands is used. This motivates us to prefer the GPF before EKF [Haykin, 2001] and other approximate filtering tools, like UKF [Wan and de Merwe, 2001] and CDF [Norgaard et al., 2000], which do not possess this property. Even in the presence of severe nonlinearities and noise in the data, the GPF achieves more accurate mean and variance estimates than other particle filters with the same number of particles [Kotecha and Djuric, 2003a].

5.2 The Gaussian Sum Particle Filter

GPF adopt Gaussian sums to represent the predictive and filtering densities, which allows us to model non-Gaussian effects. However, it has been found that they are still sensitive to large discrepancies and outliers in the data because even in these filters the state noise is assumed to be normal. A proposed solution to this problem is to introduce another mixture density to tackle the state noise [Kotecha and Djuric, 2003b]. This mixture for the state noise brings further flexibility, and when applied to modelling stochastic volatilities it helps to capture better the unknown nature of the state noise.

The GSPF [Kotecha and Djuric, 2003a, 2003b] achieves accurate approximations to the densities of interest in dynamic models by expressing the state noise ε_t as a Gaussian mixture of finite number of terms:

$$\varepsilon_t \simeq \sum_{k=1}^{K} \alpha_k \mathcal{N}\left(\varepsilon_{t,k}; \widehat{\varepsilon}_{t,k}, Q_{t,k}\right) \tag{16}$$

assuming $\widehat{\varepsilon}_{t,k}$ is the mean noise, $Q_{t,k}$ is the noise covariance matrix of the k-th mixand.

Accomodating such a special state noise model for each state in its predictive distribution leads to a complicated predictive density $p(x_t|\mathbf{y}_{0:t-1}) = \int \sum_{k=1}^{K} \alpha_k \mathcal{N}\left(x_t; h(x_{t-1}) + \widehat{\varepsilon}_{t,k}, Q_{t,k}\right) \mathcal{N}\left(x_{t-1}, m; \widehat{x}_{t-1,m}, P_{t-1,m}\right) dx_{t-1}$. This means that that the previous state x_{t-1}, from the corresponding m-th mixand with mean $\widehat{x}_{t-1,m}$, is passed through the state function $h(x_{t-1})$ and, after that, a mean noise sample from each of the k noise mixands $\widehat{\varepsilon}_{t,k}$ is added to it, thus making MK new states. The complicated expression may be reformulated so as to see that really the predictive density can be approximated by a finite Gaussian mixture:

$$\begin{aligned} \widehat{p}(x_t|\mathbf{y}_{0:t-1}) &= \sum_{m=1}^{M}\sum_{k=1}^{K} \int \alpha_k w_{t-1,m} \mathcal{N}\left(x_t; h(x_{t-1}) + \widehat{\varepsilon}_{t,k}, Q_{t,k}\right) \mathcal{N}\left(x_{t-1,m}; \widehat{x}_{t-1,m}, P_{t-1,m}\right) dx_{t-1} && (17) \\ &\simeq \sum_{i=1}^{MK} u_i \mathcal{N}\left(x_{t|t,i}; x_{t|t-1,i}, R_{t|t-1,i}\right) && (18) \end{aligned}$$

where the common weight is generated by the product $u_i = \alpha_k w_{t-1,m}$. The mean and covariance of the normal distribution are respectively $x_{t|t-1,i} = h(\widehat{x}_{t-1,m}) + \widehat{\varepsilon}_{t,k}$ and $R_{t|t-1,i} = \phi P_{t-1,m} \phi + Q_{t-1,k}$, where ϕ is the derivative of the state equation with respect to the mean state.

This stage of the filter involves several steps: 1) drawing a number $n = 1..N$ of state samples $\{x_{t-1,m}^{(n)}\}_{n=1}^{N}$ from the posterior $x_{t-1,m}^{(n)} \sim \mathcal{N}\left(x_{t-1,m}; \widehat{x}_{t-1,m}, P_{t-1,m}\right)$ of each previous mixand $m = 1..M$; 2) picking a sample $\widehat{\varepsilon}_{t,k}$ from each $k = 1..K$ noise model $\mathcal{N}\left(\varepsilon_{t,k}; \widehat{\varepsilon}_{t,k}, Q_{t,k}\right)$ and adding it to each sample using the state function $x_{t|t-1,i}^{(n)} = h(x_{t-1,m}^{(n)}) + \widehat{\varepsilon}_{t-1,k}$ (where the index $i = m + (k-1)K$ specifies a unique enumeration), thus leading to a large number NMK of particles $\{x_{t|t-1,i}^{(n)}\}_{n=1}^{N}$ from an increased number MK of mixands; 3) estimating the mean by sample averaging $\widehat{x}_{t|t-1,i} = (1/N)\sum_{n=1}^{N} x_{t|t-1,i}^{(n)}$ and the variance $R_{t|t-1,i}$ of each mixand; 4) producing weights for each $i = 1..MK$ mixand by $u_i = \alpha_k w_{t-1,m}/(\sum_{m=1}^{M}\sum_{k=1}^{K}\alpha_k w_{t-1,m})$; and 5) approximating the predictive by the following mixture:

$$\widehat{p}(x_t|\mathbf{y}_{0:t-1}) \simeq \sum_{i=1}^{MK} u_i \mathcal{N}\left(x_{t|t-1,i}; \widehat{x}_{t|t-1,i}, R_{t|t-1,i}\right) \tag{19}$$

which is a sum of increasing number of terms after each step. Since the number of mixands grows exponentially with the time, mixands with insignificant weights are removed so as to keep their number constant.

The second stage of GSPF assimilates the effect of the newly arrived observation y_t, and estimates the updated filtering density $\widehat{p}(x_t|\mathbf{y}_{0:t})$ also using Gaussian mixtures. The update of the filtering density is obtained by discrete summation:

$$\widehat{p}(x_t|y_t) \simeq C_t \sum_{i=1}^{MK} u_i \mathcal{N}\left(x_{t|t-1,i}; \widehat{x}_{t|t-1,i}, R_{t|t-1,i}\right) p\left(y_t|x_t\right) \tag{20}$$

where $p(y_t|x_t)$ is the likelihood of the model, and C_t is a normalizing constant.

This modification requires to take state samples $\{x_{t|t-1,i}^{(n)}\}_{n=1}^{N}$ from the corresponding predicted mixands $\mathcal{N}\left(x_{t|t-1,i}; \widehat{x}_{t|t-1,i}, R_{t|t-1,i}\right)$, so as to compute the importance weights. Using the state transition probabilty as proposal leads to importance sampling weights that are proportional to the model likelihood:

$$v_{t,i}^{(n)} \approx p(y_t|x_{t|t-1,i}^{(n)}) \tag{21}$$

which is estimated by passing the samples $x_{t|t-1,i}^{(n)}$ through the observation equation. Since there are generated N samples from each of the MK mixands the number of weights is NMK.

Then, the filtering density is updated using the same state samples $x_{t|t-1,i}^{(n)}$ drawn from the predictive mixture distribution. This leads to the following mean and variances computable by summation:

$$\widehat{x}_{t|t,i} = c\sum_{n=1}^{N} v_{t,i}^{(n)} x_{t|t-1,i}^{(n)} \tag{22}$$

$$R_{t|t,i} = c\sum_{n=1}^{N} v_{t,i}^{(n)} (x_{t|t-1,i}^{(n)} - \widehat{x}_{t|t,i})(x_{t|t-1,i}^{(n)} - \widehat{x}_{t|t,i})^T \tag{23}$$

where c is a normalizing constant defined by the inverse $c = \left[\sum_{n=1}^{N} v_{t,i}^{(n)}\right]^{-1}$.

The weights are calculated using the importance sampling weights $v_{t,i}^{(n)}$ of their corresponding particles:

$$\widetilde{w}_{t,i} = \frac{\sum_{n=1}^{N} v_{t,i}^{(n)}}{\sum_{i=1}^{MK} \sum_{n=1}^{N} v_{t,i}^{(n)}} \tag{24}$$

$$w_{t,i} = \frac{\widetilde{w}_{t,i}}{\sum_{i=1}^{MK} \widetilde{w}_{t,i}} \tag{25}$$

which are normalized in order to produce sensitive results.

Finally, the updated filtering density is obtained as a discrete analog of the exact posterior with the mixture:

$$\widehat{p}(x_t|\mathbf{y}_{0:t}) \simeq \sum_{i=1}^{MK} w_{t,i}\mathcal{N}\left(x_{t,i}; \widehat{x}_{t,i}, R_{t,i}\right) \tag{26}$$

which should be followed by resampling to reduce the number of mixands. Any of the resampling residual, multinomial, or systematic techniques can be applied to retain only M mixands.

The mean and covariance of this filtering density are estimated as follows:

$$\widehat{x}_{t,i} = \sum_{m=1}^{M} w_{t,i}\widehat{x}_{t|t,i} \tag{27}$$

$$R_{t,i} = \sum_{m=1}^{M} w_{t,i}\left(R_{t|t,i} + (\widehat{x}_{t,i} - \widehat{x}_{t|t,i})(\widehat{x}_{t,i} - \widehat{x}_{t|t,i})^T\right) \tag{28}$$

which take into account the selected mixtures.

It should be noted that for the studied general log volatility model (1), (2), where the input is scalar, the number of mixands is small and depends mainly on the noise model. When the noise model contains a small number of mixture components the GSPF filter for modelling the stochastic volatility with the above dynamic model do not require resampling and it is computationally fast.

6 Filtering Stochastic Volatilities

The performances of three contemporary Bayesian Monte Carlo filters, namely a sequential particle filter PF, an auxiliary particle filter APF, and a Gaussian sum particle filter GSPF, were tested on two time series: an artificially generated time series of log returns, and a real-world series of exchange rates. First, experiments were conducted to show that the GSPF has the potential to outperform the other filters on inferring the hidden volatility in an artificially generated series of returns. Second, a proper methodology for analysis of stochastic volatility models [Kim et al., 1998] was taken and followed to demonstrate how the GSPF performance relates to standard GARCH models on volatility modelling from practical financial data.

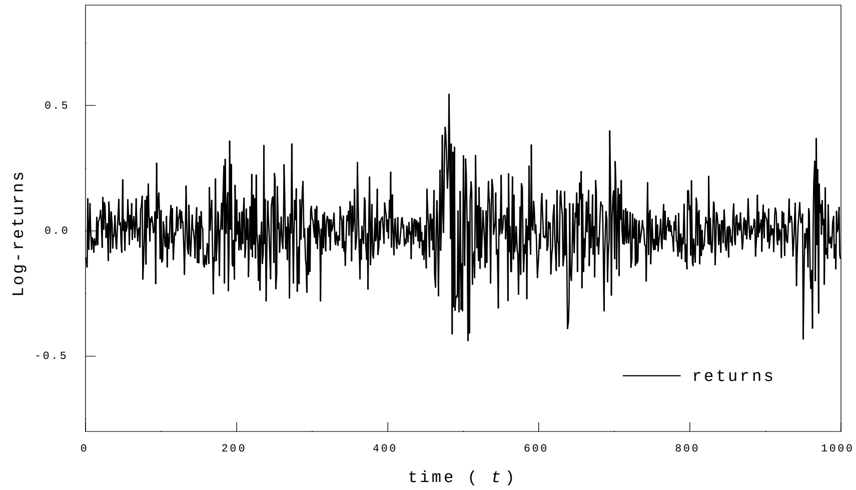

Figure 1. Log return regression targets considered for filtering.

Table 1. Comparison of the training performance of three particle filters: GSPF run with 50 particles, PF and APF run with 100 particles; each RMSE and VAR estimate is average from 20 runs.

	Training errors (volatility)					
	$1-250$		$1-500$		$1-1000$	
	$RMSE$	VAR	$RMSE$	VAR	$RMSE$	VAR
PF	0.6276	0.4562	0.6432	0.4862	0.6561	0.4986
APF	0.5845	0.2535	0.5994	0.2836	0.6022	0.2975
GSPF	0.5274	0.2163	0.5322	0.2325	0.5384	0.2451

6.1 Studies on Generated Returns

A time series was made by generating sucessively random values for the volatility x_t and passing them through the autoregressive function (1), using parameters $\gamma = log(0.01)$ and $\phi = 0.95$. This volatility was further contamined by state noise ε_t of variance $q = sqrt(0.1)$. The mean of the log returns y_t was taken to be $\mu = 0.001$. The synthesised series was divided into three segments of increased size: $250, 500$ and 1000 training points (log returns). The whole training series is illustrated in Figure 1.

Table 1 provides results obtained with PF and APF run with 100 particles, and GSPF run with 50 particles as it typically needs a smaller population. GSPF was made to use 5 mixands, for the state and filtering densities, and 2 mixands for the state noise. The results in Table 1 indicate that although all these three major kinds of sequential Monte Carlo filters have similar learning potential, the GSPF clearly outperforms them.

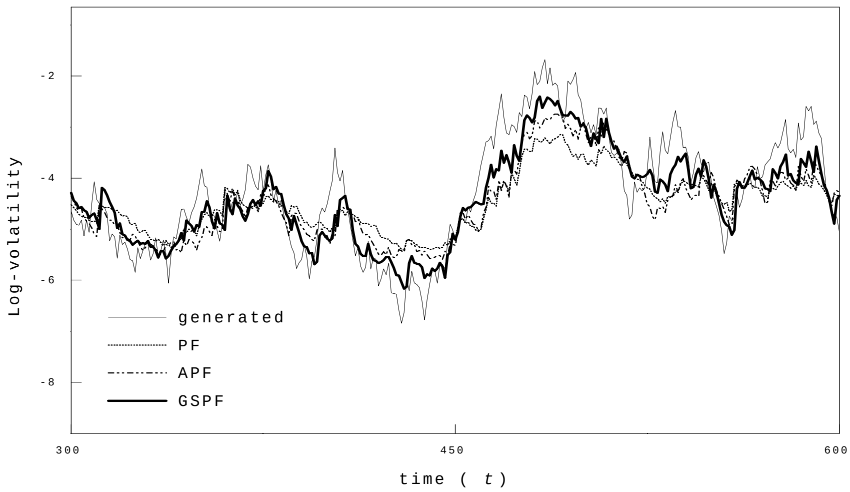

Figure 2. Sequentially filtered volatility by the Bayesian Monte Carlo algorithms, from a representative run, with curves obtained from running GSPF using 50 particles, PF and APF run using 100 particles.

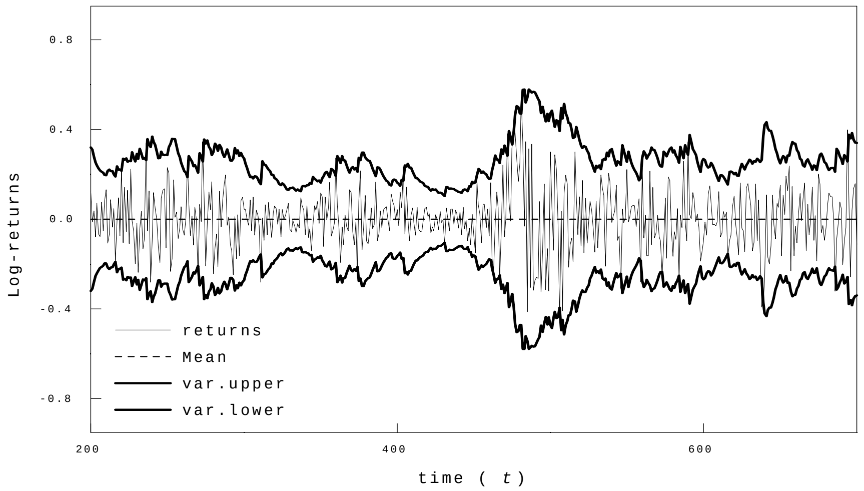

Figure 3. Learned variance of the mean return by GSPF.

In order to find out if there is a big difference in the curvatures of the identified volatilities, there were recorded the approximations by each of the filters during a representative run. The volatility curves produced by the three filters are plotted in Figure 2 along with the true (artificially generated) volatility. Figure 2 demonstrates that the volatility curve of the GSPF is much closer the the true volatility than the other curves.

The next question that we raise is whether the GSPF algorithm really works well. That is why, during the same experiments we evaluated also the inferred mean by the GSPF and the variance of this mean, which are depicted in Figure 3. The dotted line shows that the learned model has almost zero mean, whose variance wraps closely the given targets, which is an indication that GSPF works properly.

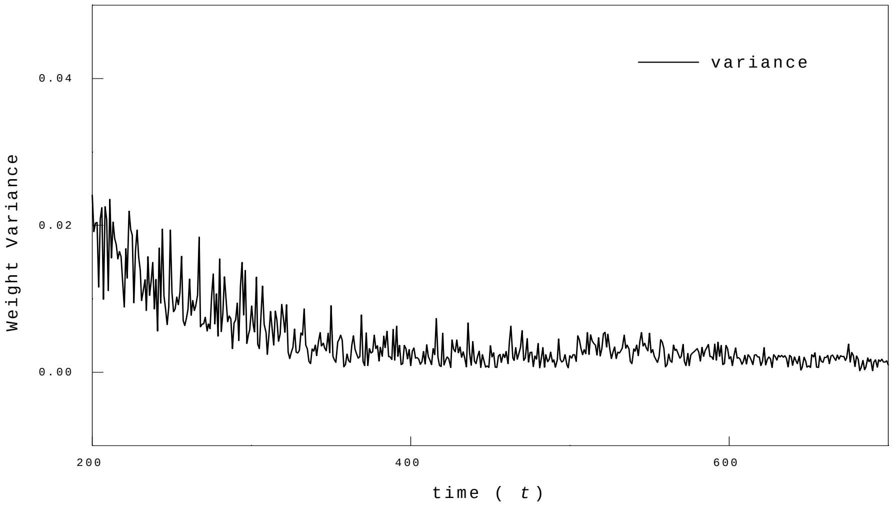

Figure 4. Weigh variance of the particles, recorded during a particular run of GSPF.

In order to be convincing about the convergence performance of the GSPF there was estimated the variance of the importance weights after each time step. Figure 4 shows that the weights variance decreases which indicates that the filter converges. Note that since the importance weights are estimated in a non-recursive manner, the weight variance of all weights in the mixture model does not decrease monotonically.

6.2 Modelling the Volatility of Exchange Rates

The proper methodology for volatility inference involves three steps [Shephard, 2005]: 1) finding the parameters, q, ϕ, and γ, of the stochastic model by sampling; 2) learning the unobserved volatility, here using the selected simulation-based filters; and 3) performing statistical diagnostics. In this subsection, we take a benchmark series of daily closing GBP/USD exchange rates from 1/10/1981 to 28/6/1985 [Kim et al., 1998]. The first 896 observations were taken for training, and the remaining 50 for testing predictability.

A Gibbs sampler was implemented and used to conduct $100,000$ sweeps over the log-differenced exchange rate series, as reccommended [Kim et al., 1998]. More precisely, the returns were pre-processed by the following formula: $y_t = 100\left[(\log r_t - \log r_{t-1}) - \sum_{i=1}^{T}(\log r_i - \log r_{i-1})/T\right]$, where r_t is the exchange rate at moment t. The initial values for the unknown parameters of the model, defined by equations (1) and (2), were considered to be: $q = 0.2$, $\phi = 0.95$, and $\beta = \exp(\gamma/2) = 0.5$ ($\eta = 1$). The inferred mean values of these parameters, obtained after an initial burn-in period of $20,000$ iterations, along with their standard deviation and cross-correlations are given in Table 2.

Table 2. Parameters of the stochastic volatility model found by Gibbs sampling after 100,000 iterations over the GBP/USD exchange rates series.

	Parameter estimates				
	Mean	*Deviation*	*Correlations of the posteriors*		
q	0.1628	0.0048	0.00074	−0.00015	−0.00034
ϕ	0.9811	0.0012	−0.00015	−0.00016	0.00021
β	0.7186	0.0032	−0.00034	0.00021	0.01015

Table 3. Statistical diagnostics of the volatilities inferred by different filters, as well as, a GARCH volatility estimate on the trainig GBP/USD exchange rates series (average results from conducted 20 independent runs).

	Volatility estimates			
	Skewness	*Kurtosis*	*LB*(30)	*Log − lik*
GARCH	2.83	3.22	12.52	−985
PF	2.54	2.92	13.84	−938
APF	2.61	2.88	19.54	−927
GSPF	1.56	2.07	14.36	−932

Next, the studied filters were run on the training log-differenced exchange rate series, using the estimated parameters from Table 2, to find the volatility at each series point. The PF and APF were run again using populations of 100 particles. The GSPF was also run again with 50 particles and 2 mixands for the state noise. There were conducted 20 independent runs with each filter, starting from random initial weights.

A standard autoregressive conditional heteroskedastic volatility model was taken for comparisons. The series was fitted by a Gaussian GARCH(1,1) [Bollerslev, 1986] model: $v_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2 + \alpha_2 v_{t-1}^2$, where: $e_{t-1} = y_{t-1} - \widehat{y}_{t-1}$, assuming $y_t = \mathcal{N}(0, v_t^2)$. Using adaptive rejection Metropolis sampling [Gilks et al., 1995], there were obtained the following parameters: $\alpha_0 = 0.001$, $\alpha_1 = 0.325$, and $\alpha_2 = 0.604$.

In order to facilitate comparisons with relevant research, there were computed the statistical measures skewness and kurtosis to obtain information for the deviation of these models from normality. There was also measured the Ljung-Box statistic [Ljung and Box, 1978] for the significance of the autocorrelations in the volatility series using 30 lags, and the log-likelihood function: $\mathcal{L} = -\log(2.0\pi x_t) - (e_t^2/x_t)$, where: $e_t = y_t - \widehat{y}_t$. These statistics, calculated over the training series and averaged over 20 runs, are given in Table 3.

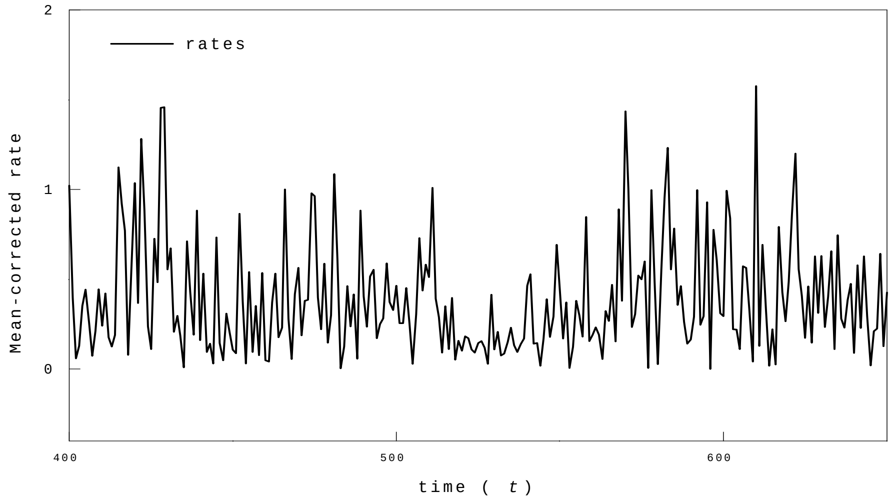

Figure 5. A segment from the log-differenced series of closing daily GBP/USD exchange rates.

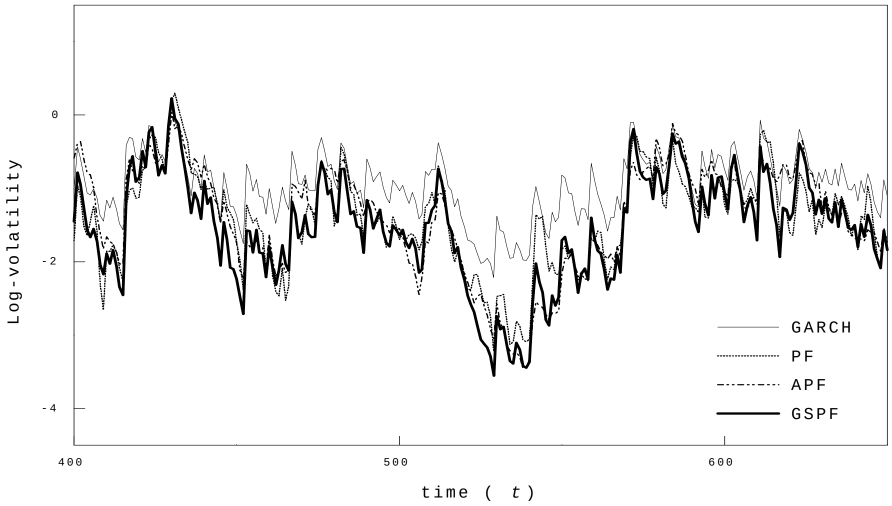

Figure 6. Volatility curves, obtained in particular runs, with each of the studied filters and the GARCH model.

The results is Table 3 show that the GSPF results feature smallest deviation from normality, low autocorrelations, and high likelihood. Highest likelihood has the result from APF but it has higher autocorrelation, as well as higher skewness and kurtosis. Although the GARCH model has lowest LB(30) etimate, it features lowest likelihood. All particle filters lead to models with high likelihood which is an effect not only from the Monte Carlo simulation procedure, but it is also due to resampling. The resampling, however, cases instabilities in learning, and it is not surprising that the GSPF filter which does not resample performs best.

Figure 5 shows the approximation of a segment from the training GBP/USD exchange rates series. The inferred volatilities from the same time interval are plotted in Figure 6. It can be seen that the changes in the volatility reflect the degree of fluctuation in the returns.

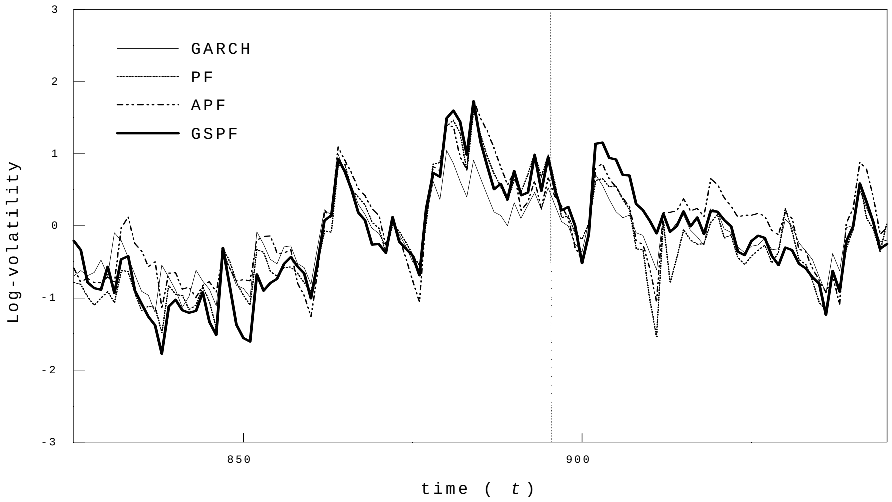

Figure 7. One-step ahead forecasts of the volatility of the closing daily GBP/USD exchange rates, obtained in particular runs, with each of the studied algorithms.

Table 4. Statistical diagnostics of the one-step ahead volatility forecasts over a future, unseen testing period of the GBP/USD exchange rates series.

	Volatility estimates			
	Skewness	*Kurtosis*	$LB(30)$	$Log - lik$
GARCH	0.23	0.34	7.64	-145
PF	0.81	0.92	18.87	-75
APF	0.53	0.67	22.13	-71
GSPF	0.55	0.83	9.51	-73

Table 4 gives the experimental results from one-step ahead forecasting of the last 50 points ($t = 897$ and $t = 946$) from the GBP/USD exchange rates series. It should be noted that the GARCH model although being closest to normality, in comparison with the particle filters, exhibits again much lower likelihood. The APF filter shows highest likelihood, but highest autocorrelation between the future points. The GSPF filter seems to have stable performance with good statictical characteristics.

Figure 7 provides the plots of the predicted volatility at each point from the future interval.

7 Conclusion

This paper presented mixture Monte Carlo particle filters as a sophisticated contemporary approach that is suitable for stochastic volatility modelling. Empirical investigations show that the Gaussian sum particle filter from this class outperforms even the benchmark Auxiliary Particle Filter on volatility accuracy and variance. An important advantage of GSPF is that it attains good results with acceptable computational speed, due to the need to maintain a smaller number of particles.

The Gaussian mixture particle filters could be of interest for practitioners in various areas, such as asset pricing, risk management and portfolio optimization.

References

[1] Anderson, B.D.O. and Moore, J.B. (1979). *Optimal Filtering*, Prentice Hall Inc., Englewood Cliffs, NJ.

[2] Bauwens, L., Lubrano, M. and Richard, J.F. (1999). *Bayesian Inference in Dynamic Econometric Models*, Oxford University Press, New York.

[3] Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity, *Journal of Econometrics*, vol.31, pp.307-327.

[4] Carpenter, J., Clifford, P. and Fearnhead, P. (1999). An Improved Particle Filter for Non-linear Problems, *IEE Proceedings- Radar, Sonar and Navigation*, vol.146, pp.2-7.

[5] Chen, R. and Liu, J. (2000). Mixture Kalman Filters, *Journal of the Royal Statistical Society B*, vol.62, N:3, pp.493-508.

[6] de Freitas, J.F.G., Niranjan, M.A., Gee, A.H. and Doucet, A. (2000). Sequential Monte Carlo Methods to Train Neural Network Models, *Neural Computation*, vol.12, N:4, pp.955-993.

[7] Doucet, A., de Freitas, N. and Gordon, N. (Eds). (2001). *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York.

[8] Durbin, J. and Koopman, S.J. (2000). Time Series Analysis of Non-Gaussian Observations Based on State Space Models from Both Classical and Bayesian Perespective, *Journal of the Royal Statistical Society B*, vol.62, N:1, pp.3-156.

[9] Geweke, J. (1989). Bayesian Inference in Econometric Models using Monte Carlo Integration, *Econometrica*, vol.24, pp.1317-1399.

[10] Ghysels, E., Harvey, A.C. and Renault, E. (1996). Stochastic volatility, in: C.R.Rao and G.S.Maddala (Eds.), *Statistical Methods in Finance*, North-Holland, Amsterdam, pp.119-191.

[11] Gilks, W.R., Best, N.G. and Tan, K.K.C. (1995). Adaptive Rejection Metropolis Sampling within Gibbs Sampling, *Applied Statistics*, vol.44, N:4, pp.455-472.

[12] Gordon, N.J., Salmond, D.J. and Smith, A.F.M. (1993). Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation, *Proceedings IEE-F*, vol.140, pp.107-113.

[13] Harvey, A.C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge.

[14] Harvey, A.C., Ruiz, E. and Shephard, N. (1994). Multivariate Stochastic Variance Models, *Rview of Economic Studies*, vol.61, N:2, pp.247-264.

[15] Haykin, S. (Ed.) (2001). *Kalman Filtering and Neural Networks*, John Wiley and Sons, New York, NY.

[16] Hull, J. and White, A. (1987). The Pricing of Options on Assets with Stochastic Volatilities, *The Journal of Finance*, vol.42, N:2, pp.281-300.

[17] Hull, J.C. (2005). *Options, Futures and Other Derivatives*, 6th ed., Prentice Hall, New Jersey.

[18] Jacquier, E., Polson, N.G. and Rossi, P.E. (2002). Bayesian Analysis of Stochastic Volatility Models, *Journal of Business and Economic Statistics*, vol.20, N:1, pp.69-87.

[19] Javaheri, A., Lautier, D., and Galli, A. (2003). Filtering in Finance, *Wilmott Magazine*, N:5.

[20] Kim, s., Shephard, N. and Chib, S. (1998). Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models, *Review of Economic Studies*, vol.65, N:3, pp.361-393.

[21] Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, *Journal of Computational and Graphical Statistics*, vol.5, N:1, pp.1-25.

[22] Kotecha, J.H. and Djuric, P.M. (2003a). Gaussian Particle Filtering, *IEEE Trans. on Signal Processing*, vol.51, N:10, pp.2592-2601.

[23] Kotecha, J.H. and Djuric, P.M. (2003b). Gaussian Sum Particle Filtering, *IEEE Trans. on Signal Processing*, vol.51, N:10, pp.2602-2612.

[24] Liu, J.S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems, *Journal of the American Statistical Society*, vol.93, N:443, pp.1032-1044.

[25] Ljung, G.M. and G.E.P. Box (1978). On a Measure of Lack of Fit in Time Series Models, *Biometrika*, vol.65, pp.553-564.

[26] Maskell, S. (2004). An Introduction to Particle Filters, In: A.Harvey, S.J.Koopman and N.Shephard (Eds.) *State Space and Unobserved Component Models*, Cambridge University Press, Cambridge, pp.40-71.

[27] Nikolaev, N. and Iba, H. (2006). *Adaptive Learning of Polynomial Networks: Genetic Programming, Backpropagation and Bayesian Methods*, Springer, New York.

[28] Norgaard, M., Poulsen, N.K. and Ravn, O. (2000). New Developments in State Estimation for Nonlinear Systems, *Automatica*, vol.36, N:11, pp. 1627-1638.

[29] Pitt, M.K. and Shephard, N. (1999). Filtering via Simulation: Auxiliary Particle Filters, *Journal of the American Statistical Association*, vol.94, N:446, pp.590-599.

[30] Poon, s.-H. and Granger, C.W.J. (2003). Forecasting Volatility in financial Markets: A Review, *Journal of Economic Literature*, vol.XLI, pp.478-539.

[31] Shephard, N. (1993). Fitting Non-linear Time Series Models, with Applications to Stochastic Variance Models, *Journal of Appled Econometrics*, vol.8, S135-S152.

[32] Shephard, N. (Ed.) (2005). *Stochastic Volatility: Selected Readings*, Advanced Texts in Econometrics, Oxford University Press, New York.

[33] Stroud, R., Polson, N.G. and Müller, P. (2004). Practicle Filtering for Stochastic Volatility Models, In: A.Harvey, S.J.Koopman and N.Shephard (Eds.) *State Space and Unobserved Component Models*, Cambridge University Press, Cambridge, pp.236-247.

[34] Wan, E.A. and van der Merwe, R. (2001). The Unscented Kalman Filter, chapter 7, in: S.Haykin (Ed.), *Kalman Filtering and Neural Networks*, Wiley, New Yrok, pp.221-269.

[35] Zoeter, O., Ypma, A. and and Heskes, T. (2004). Improved Unscented Kalman Smoothing for Stock Volatility Estimation, in: A.Barros, J.Principe, J.Larsen, T.Adali, and S.Douglas (Eds.), *Machine Learning for Signal Processing, Proc. of the 14th IEEE Signal Processing Society Workshop*, IEEE Press, NJ, pp.143- 152.