

Fičura, Milan; Witzany, Jiří

Article

Use of adapted particle filters in SVJD models

European Financial and Accounting Journal

Provided in Cooperation with:

Faculty of Finance and Accounting, University of Economics, Prague

Suggested Citation: Fičura, Milan; Witzany, Jiří (2018) : Use of adapted particle filters in SVJD models, European Financial and Accounting Journal, ISSN 1805-4846, University of Economics, Faculty of Finance and Accounting, Prague, Vol. 13, Iss. 3, pp. 5-20, <https://doi.org/10.18267/j.efaj.211>

This Version is available at:

<https://hdl.handle.net/10419/242257>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Use of Adapted Particle Filters in SVJD Models

Milan Fičura – Jiří Witzany *

Abstract:

Particle Filter algorithms for filtering latent states (volatility and jumps) of Stochastic-Volatility Jump-Diffusion (SVJD) models are being explained. Three versions of the SIR particle filter with adapted proposal distributions to the jump occurrences, jump sizes, and both are derived and their performance is compared in a simulation study to the un-adapted particle filter. The filter adapted to both the jump occurrences and jump sizes achieves the best performance, followed in their respective order by the filter adapted only to the jump occurrences and the filter adapted only to the jump sizes. All adapted particle filters outperformed the un-adapted particle filter.

Key words: Particle Filters; Stochastic Volatility; Price Jumps.

JEL classification: C11, C14, C15, C22, G1.

1 Introduction

Modelling of volatility and jumps in financial time series plays an important role in many areas of finance, such as asset pricing, portfolio optimisation, VaR estimation, option valuation or quantitative trading. Stochastic-Volatility Jump-Diffusion (SVJD) models (Bates, 1996) treat volatility and jumps as latent (unobservable) stochastic processes. This allows them to incorporate wide variety of empirically observed features of financial time series dynamics, such as volatility clustering, long memory, leverage effect, volatility jumps and jump clustering. Estimation of these models is, however, notoriously difficult (see Shephard, 2005). Unlike models that treat volatility as a deterministic function of past asset price returns (such as GARCH), use of SVJD models involves estimation of unobservable trajectories of the latent state variables (volatility and jumps). This cannot be easily done with standard inference methods such as maximum likelihood estimation (MLE) or generalised method of moments

* Milan Fičura; University of Economics, Prague, Faculty of Finance and Accounting, Department of Banking and Finance, W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic, <milan.ficura@vse.cz>.

Jiří Witzany; University of Economics, Prague, Faculty of Finance and Accounting, Department of Banking and Finance, W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic, <jiri.witzany@vse.cz>.

The article is processed as an output of the research project Advanced methods of risk management with the use of artificial intelligence VŠE IGS F1/59/2017, and the research project GAČR 18-05244S, Innovation Approaches to Credit Risk Management

(GMM), and computationally intensive, simulation-based approaches must be used instead (Andersen et al., 2005).

Among methods typically used for SVJD model estimation are the Efficient Method of Moments and the Markov–Chain Monte Carlo (MCMC) method (Jacquier et al., 2007). MCMC has become one of the most popular approaches used for SVJD model estimation (Eraker et al., 2003, Witzany, 2013, Fičura and Witzany, 2017). A particular drawback of using MCMC in tasks involving volatility and jumps forecasting is that the latent-state time series need to be re-estimated whenever new data arrive, which can be very time consuming. A possible solution to this problem is to combine MCMC with Particle Filters, allowing us to filter the evolution of the latent states sequentially as new data arrive. The most commonly used approach is to estimate the model parameters and latent states with MCMC on the in-sample period and then use the Particle Filter to filter the evolution of the latent states during the out-sample period. Another approach is to use Particle Filters not only for the latent states filtering, but also for the sequential learning of model parameters (see Fulop and Li 2013, and Fulop, Li and Yu, 2015).

The goal of the presented study is to derive adapted Particle Filters for filtering of latent states in SVJD models with self-exciting jumps. While the un-adapted SIR particle filter (Gordon et al. 1993) represents a universal filtering approach, its sampling efficiency may be low (Pitt and Shephard, 1999), which turns out to be particularly problematic when it is applied for the filtering of SVJD model jumps, especially if they are rare. To cope with this problem, we derive three possible adaptation schemes, adapting the SIR particle filter either to the jump occurrences, jump sizes, or both, and we show in a simulation study that these jump-adapted particle filters dramatically outperform the un-adapted particle filter.

The rest of this study is organised as follows. In section two, the filtering problem is described and the basic un-adapted SIR particle filter is explained. In section three, the SVJD model with self-exciting jumps is introduced and adapted particle filters are discussed. In the following three sections, we derive three jump-adapted particle filters which are subsequently applied in the simulation study and their performance is compared with the un-adapted particle filter. In the final section, we conclude the results and discuss possible areas for future research.

2 SIR Particle Filter

The goal of the latent states filtering is to estimate the evolution of latent state time series x_t from the evolution of an observable time series y_t that is somehow dependent on x_t but is plagued by observation noise.

A general state-space model can be written as follows:

$$y_t = H(x_t, w_t, \theta), \quad (1)$$

$$x_t = F(x_{t-1}, v_t, \theta), \quad (2)$$

where y_t is the observation time series, conditionally independent given the hidden state x_t . $H(\cdot)$ and $F(\cdot)$ are model functions, θ is the vector of model parameters and w_t and v_t are noise terms.

We further define the *observation density* as:

$$p(y_t | x_t, \theta) \quad (3)$$

and the *transition density* as:

$$p(x_t | x_{t-1}, \theta) \quad (4)$$

with the hidden state x_t following the Markov process with initial density $p(x_0 | \theta)$.

The joint task of latent state filtering and parameter learning is to find the joint posterior density of the hidden states and model parameters, based on all currently available information at time t . The problem can be divided into:

$$p(x_t, \theta | y_{1:t}) = p(x_t | y_{1:t}, \theta) p(\theta | y_{1:t}), \quad (5)$$

where the density $p(x_t | y_{1:t}, \theta)$ solves the *filtering problem* and $p(\theta | y_{1:t})$ solves the *parameter learning problem*.

When the functions $H(\cdot)$ and $F(\cdot)$ are linear and the distributions of x_0 , w_t and v_t are Gaussian, the problem can be solved analytically using the Kalman Filter.

In the general non-linear and non-Gaussian setting, Particle Filters can be used, approximating the posterior density $p(x_t, \theta | y_{1:t})$ with a set of weighted particles.

In the rest of the text, we will focus on the filtering problem of estimating $p(x_t | y_{1:t}, \theta)$ for the cases where the parameters θ are known and we will thus omit θ from the notation. In practical settings, the parameter vector θ has to be estimated, either on the in-sample period with MCMC or EMM, or in a sequential way with Particle Learning (see Fulop and Li 2013 and Fulop, Li and Yu, 2015).

When θ is known, it is possible to estimate the densities $p(x_t | y_{1:t}, \theta)$ for all t with the Sequential Importance Re-Sampling (SIR) Particle Filter (Gordon et al., 1993).

To employ the filter, we first need an estimate of the latent state density $p(x_0)$ at time $t = 0$, represented as a set of M weighted particles. This initial distribution is usually drawn from some wide enough prior distribution of the latent states.

For all further steps $t = 1, \dots, T$ the particle filter proceeds as follows.

Given M particles $\{x_{t-1}^{(i)}; i = 1, 2, \dots, M\}$ with weights $\tilde{w}_{t-1}^{(i)}$, representing the density $p(x_{t-1}|y_{1:t-1})$ at time $t - 1$, we use the following recursion:

$$p(x_t|y_{1:t}) \propto \int p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}, \quad (6)$$

where \propto means that the density $p(x_t|y_{1:t})$ is proportional to the right-hand side of the equation up to a normalising constant.

To approximate $p(x_t|y_{1:t})$ we draw random particles (i.e. proposal values of the latent states) from a known proposal density $g(x_t|x_{t-1})$ and assign them with importance weights given as:

$$w_t^{(i)} = \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{g(x_t^i|x_{t-1}^i, y_t)}\tilde{w}_{t-1}^{(i)} \quad (7)$$

In the most universal, un-adapted version of the particle filter, the proposal density is equal to the transition density:

$$g(x_t^i|x_{t-1}^i, y_t) = p(x_t^i|x_{t-1}^i) \quad (8)$$

And the weight update is thus equal to:

$$w_t^{(i)} = p(y_t|x_t^i)\tilde{w}_{t-1}^{(i)}, \quad (9)$$

where $p(y_t|x_t^i)$ is the likelihood of the observation y_t given the latent state x_t^i .

Normalized weights for time t are then computed as:

$$\tilde{w}_t^{(i)} = w_t^{(i)} / \sum_{j=1}^M w_t^{(j)} \quad (10)$$

Arriving at a weighted set of particles $\{x_t^{(i)}; i = 1, 2, \dots, M\}$ with weights $\tilde{w}_t^{(i)}$, representing the density $p(x_t|y_{1:t})$ at time t .

As the particle weights tend to degenerate after few iterations to the state where only one particle has a non-negligible weight, a re-sampling step is usually added to the algorithm. In the re-sampling step, the particles are re-sampled with a replacement from the current particles set with probabilities proportional to the particle weights.

The re-sampling can be performed after each iteration or only when the efficient number of particles (ESS) drops below a certain threshold (ESS_{Thr}):

$$ESS = 1 / \sum_{j=1}^M (\tilde{w}_t^{(j)})^2 < ESS_{Thr} \quad (11)$$

3 SVJD Model and the Un-adapted Particle Filter

In the presented study, the SIR Particle Filter is applied for filtering of latent states of the SVJD model with self-exciting jumps. The SVJD model has 3 equations.

The first equation describes the process of logarithmic asset price returns as:

$$r_t = \mu + \sigma_t \varepsilon_t + J_t Q_t, \quad (12)$$

where $r_t = p_t - p_{t-1}$ is the logarithmic return, p_t is the logarithm of the asset price, μ is the drift rate, σ_t is the stochastic volatility, $\varepsilon_t \sim N(0,1)$ is a standard normal white noise, $J_t \sim N(\mu_j, \sigma_j)$ is a latent variable determining the jump sizes, and $Q_t \sim \text{Bern}[\lambda_t]$ is a latent variable determining the jump occurrences, so that $\Pr[Q_t = 1] = \lambda_t$, and λ_t is the latent jump intensity.

The second equation describes the process of the log-variance as:

$$h_t = \alpha + \beta h_{t-1} + \gamma \varepsilon_{V,t}, \quad (13)$$

where $h_t = \log(V_t)$ is the logarithm of the stochastic variance $V_t = \sigma_t^2$, α is the constant of the model, determining the long-term log-variance via the relationship $\alpha = (1 - \beta)h_{LT}$, β is the auto-regression coefficient, γ is the volatility of the log-variance, and $\varepsilon_{V,t} \sim N(0,1)$ is Gaussian white noise in the log-variance process.

The third equation corresponds to the discretized Hawkes process, governing the self-exciting behaviour of jumps, by defining the process of the jump intensity as:

$$\lambda_t = \alpha_j + \beta_j \lambda_{t-1} + \gamma_j Q_{t-1}, \quad (14)$$

where λ_t is the jump intensity, α_j determines the long-term jump intensity via $\alpha_j = (1 - \beta_j - \gamma_j)\lambda_{LT}$, β_j is the speed of the exponential decay of the jump intensity towards its long-term level λ_{LT} , and γ_j is the self-exciting parameter, determining how much will the jump intensity increase within a day following the jump Q_{t-1} .

The SVJD model contains 3 series of latent state variables. Stochastic variances V_t , jump occurrences Q_t and jump sizes J_t . These latent time series represent the vector latent state variable x_t in the particle filter notation, while the observed logarithmic returns represent the observable variable y_t .

As already mentioned, in the un-adapted particle filter, the proposal density equals the transition density $g(x_t^i | x_{t-1}^i, y_t) = p(x_t^i | x_{t-1}^i)$. In the SVJD model, the proposal density will thus be:

$$p(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i) \quad (15)$$

and the proposal values of h_t^i, J_t^i, Q_t^i can be sampled independently from:

$$p(h_t^i | h_{t-1}^i) \sim N(\alpha + \beta h_{t-1}, \gamma) \quad (16)$$

$$p(J_t^i) \sim N(\mu_J, \sigma_J) \quad (17)$$

$$p(Q_t^i | \lambda_t^i) \sim \text{Bern}[\lambda_t^i] \quad (18)$$

The likelihood function of the model is:

$$p(r_t | h_t^i, J_t^i, Q_t^i) \sim N(r_t; \mu + J_t^i Q_t^i, \sigma_t^i) \quad (19)$$

And the weight update will be:

$$w_t^{(i)} = p(r_t | h_t^i, J_t^i, Q_t^i) \tilde{w}_{t-1}^{(i)} \quad (20)$$

4 Adapted Particle Filters

As the proposal density of the un-adapted particle filter does not utilise the information about the observation y_t (which is already known at t), it is sub-optimal and it tends to propose many latent state values that end up being discarded during the weight update and re-sampling step. This inefficiency is especially severe in the case of jumps, as the value of λ_t is often very small and the vast majority of latent state proposals will thus give $Q_t^i = 0$, in spite of the fact that the corresponding r_t may be very large and thus indicative of the jump. Even the few proposals that happen to give $Q_t^i = 1$ may subsequently also get discarded, as the value of the independently sampled J_t^i may be inadequate or even have the opposite sign to the observed return. The un-adapted sampling of proposals may thus be very inefficient with respect to the jumps, and a large number of proposals will thus have to be generated to achieve reasonable accuracy of the particle filter (for discussion of alternatives, such as the Auxiliary Particle Filter, see Pitt and Shephard, 1999).

Solution to his problem is to use an adapted filter, utilising the information about r_t during sampling of the proposals. In the ideal case of a fully adapted particle filter, the proposal density would look as follows:

$$p(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \quad (21)$$

Unfortunately, this density is of an unknown form for the analysed SVJD model, due to the stochastic variance term, that it is log-normal with respect to V_{t-1}^i , but inverse-gamma with respect to r_t . For approximations of how to sample V_t^i in the MCMC setting which could potentially also be applied in particle filters, see Jacquier et al. (1994), or Kim, Shephard and Chib (1998).

Fortunately, employment of the particle filter does not require fully adapted proposal density and an approximate adaptation may be used instead.

Denoting the approximately adapted proposal distribution as:

$$g(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \quad (22)$$

we just need to appropriately assign the updated weights to the sample:

$$w_t^{(i)} = \frac{p(r_t | h_t^i, J_t^i, Q_t^i) p(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t)}{g(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t)} \tilde{w}_{t-1}^{(i)} \quad (23)$$

in order to get the weighted sample from $p(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t)$.

As sampling of the asset price jumps is the least efficient part of the sampling, we proceed to derive adapted proposal distributions with respect to the jump occurrences (Q_t^i) and jump sizes (J_t^i), which can be done analytically, while leaving the sampling of the stochastic variances un-adapted.

5 Particle Filter Adapted to Jump Sizes

To adapt the proposal density to the jump sizes J_t^i , we rewrite it as follows:

$$\begin{aligned} &g(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \\ &= p(J_t^i | h_t^i, Q_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) p(h_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i) \end{aligned} \quad (24)$$

We then sample from $p(h_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i)$ the proposals of h_t^i and Q_t^i independently, using the transition densities:

$$p(h_t^i | h_{t-1}^i) \sim N(\alpha + \beta h_{t-1}^i, \gamma) \quad (25)$$

$$p(Q_t^i | \lambda_t^i) \sim \text{Bern}[\lambda_t^i] \quad (26)$$

In the second step, we sample from the adapted density:

$$p(J_t^i | h_t^i, Q_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \quad (27)$$

which is equivalent to $p(J_t^i | h_t^i, r_t, Q_t^i)$, and we can sample from it as follows.

If $Q_t^i = 0$, then $p(J_t^i | h_t^i, r_t, Q_t^i = 0) = p(J_t^i)$ and we sample from:

$$p(J_t^i) \sim N(\mu_J, \sigma_J) \quad (28)$$

If $Q_t^i = 1$, then we can express $p(J_t^i | h_t^i, r_t, Q_t^i = 1)$ with the Bayes theorem as:

$$p(J_t^i | h_t^i, r_t, Q_t^i = 1) \propto p(r_t | h_t^i, J_t^i, Q_t^i) p(J_t^i) \quad (29)$$

As both right hand side densities are Gaussian in J_t^i :

$$p(J_t^i | h_t^i, r_t, Q_t^i = 1) \propto N(r_t; \mu + J_t^i, \sigma_t^i) N(J_t^i; \mu_J, \sigma_J) \quad (30)$$

we can derive the adapted Gaussian density to sample from as:

$$p(J_t^i | h_t^i, r_t, Q_t^i = 1) \sim N(\mu_j^*, \sigma_j^*) \quad (31)$$

$$\mu_j^* = \frac{(r_t - \mu)\sigma_j^2 + \mu_j V_t^i}{\sigma_j^2 + V_t^i} \quad (32)$$

$$\sigma_j^* = \frac{\sigma_j \sigma_t^i}{\sqrt{\sigma_j^2 + V_t^i}} \quad (33)$$

and the importance weights update is:

$$w_t^{(i)} = \frac{p(r_t | h_t^i, J_t^i, Q_t^i) p(J_t^i)}{p(J_t^i | h_t^i, r_t, Q_t^i)} \tilde{w}_{t-1}^{(i)} \quad (34)$$

6 Particle Filter Adapted to Jump Occurrences

To adapt the proposal density with respect to the jump occurrences Q_t^i , we rewrite the proposal density as:

$$\begin{aligned} g(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \\ = p(Q_t^i | h_t^i, J_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) p(h_t^i, J_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i) \end{aligned} \quad (35)$$

We then sample from $p(h_t^i, J_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i)$, the values of h_t^i and J_t^i , independently, by using the transition densities:

$$p(h_t^i | h_{t-1}^i) \sim N(\alpha + \beta h_{t-1}^i, \gamma) \quad (36)$$

$$p(J_t^i) \sim N(\mu_j, \sigma_j) \quad (37)$$

Then we sample Q_t^i from:

$$p(Q_t^i | h_t^i, J_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \quad (38)$$

which is in our case equal to $p(Q_t^i | h_t^i, J_t^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t)$ and $p(Q_t^i | h_t^i, J_t^i, \lambda_t^i, r_t)$, as the jump intensity λ_t^i is given deterministically by Q_{t-1}^i and λ_{t-1}^i .

From the Bayes theorem we can derive the adapted density as:

$$p(Q_t^i | h_t^i, J_t^i, \lambda_t^i, r_t) \sim \text{Bern}[\lambda_t^{i*}] \quad (39)$$

$$\lambda_t^{i*} = \frac{N(r_t; \mu + J_t^i, \sigma_t^i) \lambda_t^i}{N(r_t; \mu + J_t^i, \sigma_t^i) \lambda_t^i + N(r_t; \mu, \sigma_t^i) (1 - \lambda_t^i)} \quad (40)$$

And the importance weight update is:

$$w_t^{(i)} = \frac{p(r_t|h_t^i, J_t^i, Q_t^i)(\lambda_t^i)^{Q_t^i}(1-\lambda_t^i)^{1-Q_t^i}}{(\lambda_t^{i*})^{Q_t^i}(1-\lambda_t^{i*})^{1-Q_t^i}} \tilde{w}_{t-1}^{(i)} \quad (41)$$

7 Particle Filter Adapted to Jump Sizes and Occurrences

We can rewrite the proposal density as:

$$\begin{aligned} g(h_t^i, J_t^i, Q_t^i|h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \\ = p(J_t^i, Q_t^i|h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t)p(h_t^i|h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i) \end{aligned} \quad (42)$$

And decompose the bi-variate density for J_t^i and Q_t^i into:

$$\begin{aligned} p(J_t^i, Q_t^i|h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \\ = p(J_t^i|h_t^i, Q_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t)p(Q_t^i|h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) \end{aligned} \quad (43)$$

which equivalent to:

$$p(J_t^i, Q_t^i|h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t) = p(J_t^i|h_t^i, Q_t^i, r_t)p(Q_t^i|h_t^i, \lambda_t^i, r_t) \quad (44)$$

for $Q_t^i = 0$ it holds that:

$$p(J_t^i|h_t^i, r_t, Q_t^i = 0) = p(J_t^i) \quad (45)$$

with $p(J_t^i) \sim N(\mu_J, \sigma_J)$.

For $Q_t^i = 1$ we need to use the Bayes theorem to express $p(J_t^i|h_t^i, r_t, Q_t^i = 1)$ as:

$$p(J_t^i|h_t^i, r_t, Q_t^i = 1) \propto p(r_t|h_t^i, J_t^i, Q_t^i)p(J_t^i) \quad (46)$$

which is a multiple of two Gaussian densities:

$$p(J_t^i|h_t^i, r_t, Q_t^i = 1) \propto N(r_t; \mu + J_t^i, \sigma_t^i)N(J_t^i; \mu_J, \sigma_J) \quad (47)$$

and the adapted density can be derived as:

$$p(J_t^i|h_t^i, r_t, Q_t^i = 1) \sim N(\mu_J^*, \sigma_J^*) \quad (48)$$

$$\mu_J^* = \frac{(r_t - \mu)\sigma_J^2 + \mu_J V_t^i}{\sigma_J^2 + V_t^i} \quad (49)$$

$$\sigma_J^* = \frac{\sigma_J \sigma_t^i}{\sqrt{\sigma_J^2 + V_t^i}} \quad (50)$$

The next task is to derive the density $p(Q_t^i|h_t^i, \lambda_t^i, r_t)$. Using the Bayes theorem:

$$p(Q_t^i|h_t^i, \lambda_t^i, r_t) \propto p(r_t|h_t^i, Q_t^i)p(Q_t^i|\lambda_t^i) \quad (51)$$

the density $p(r_t|h_t^i, Q_t^i)$ is the marginal of $p(r_t, J_t|h_t^i, Q_t^i)$, integrated over J_t .

When $Q_t^i = 0$, then J_t and r_t are independent, and thus:

$$p(r_t|h_t^i, Q_t^i = 0) = N(r_t; \mu, \sigma_t^i) \quad (52)$$

When $Q_t^i = 1$, then we decompose the density $p(r_t, J_t|h_t^i, Q_t^i = 1)$ into:

$$p(r_t, J_t|h_t^i, Q_t^i = 1) = p(r_t|J_t, h_t^i, Q_t^i)p(J_t|h_t^i, Q_t^i) \quad (53)$$

Which is a multiple of two Gaussian densities:

$$p(r_t, J_t|h_t^i, Q_t^i = 1) = N(r_t; \mu + J_t, \sigma_t^i)N(J_t; \mu_J, \sigma_J) \quad (54)$$

Or equivalently:

$$p(r_t, J_t|h_t^i, Q_t^i = 1) = N(r_t - \mu; J_t, \sigma_t^i)N(J_t; \mu_J, \sigma_J) \quad (55)$$

The density $r_t - \mu$ is a Compound Gaussian density with mean distributed according to the Gaussian density $N(J_t; \mu_J, \sigma_J)$. We can thus integrate over J_t to get:

$$p(r_t|h_t^i, Q_t^i = 1) = N\left(r_t; \mu_J + \mu, \sqrt{\sigma_J^2 + V_t^i}\right) \quad (56)$$

Returning to the relationship $p(Q_t^i|h_t^i, \lambda_t^i, r_t) \propto p(r_t|h_t^i, Q_t^i)p(Q_t^i|\lambda_t^i)$, we see that we can easily compute the normalising constant, as the Q_t^i is only binary.

Thus, we get for $p(Q_t^i|h_t^i, \lambda_t^i, r_t)$:

$$p(Q_t^i|h_t^i, \lambda_t^i, r_t) \sim \text{Bern}[\lambda_t^{i*}], \quad (57)$$

$$\lambda_t^{i*} = \frac{N\left(r_t; \mu_J + \mu, \sqrt{\sigma_J^2 + V_t^i}\right) \lambda_t^i}{N\left(r_t; \mu_J + \mu, \sqrt{\sigma_J^2 + V_t^i}\right) \lambda_t^i + N(r_t; \mu, \sigma_t^i)(1 - \lambda_t^i)} \quad (58)$$

In order to sample from the proposal density, we will first sample from the transition density $p(h_t^i|h_{t-1}^i)$, given as:

$$p(h_t^i|h_{t-1}^i) \sim N(\alpha + \beta h_{t-1}^i, \gamma) \quad (59)$$

Then we sample from the adapted jump occurrence density $p(Q_t^i | h_t^i, \lambda_t^i, r_t)$:

$$p(Q_t^i | h_t^i, \lambda_t^i, r_t) \sim \text{Bern}[\lambda_t^{i*}] \quad (60)$$

And finally, the adapted jump sizes from $p(J_t^i | h_t^i, Q_t^i, r_t)$, as already described.

The importance weight update is:

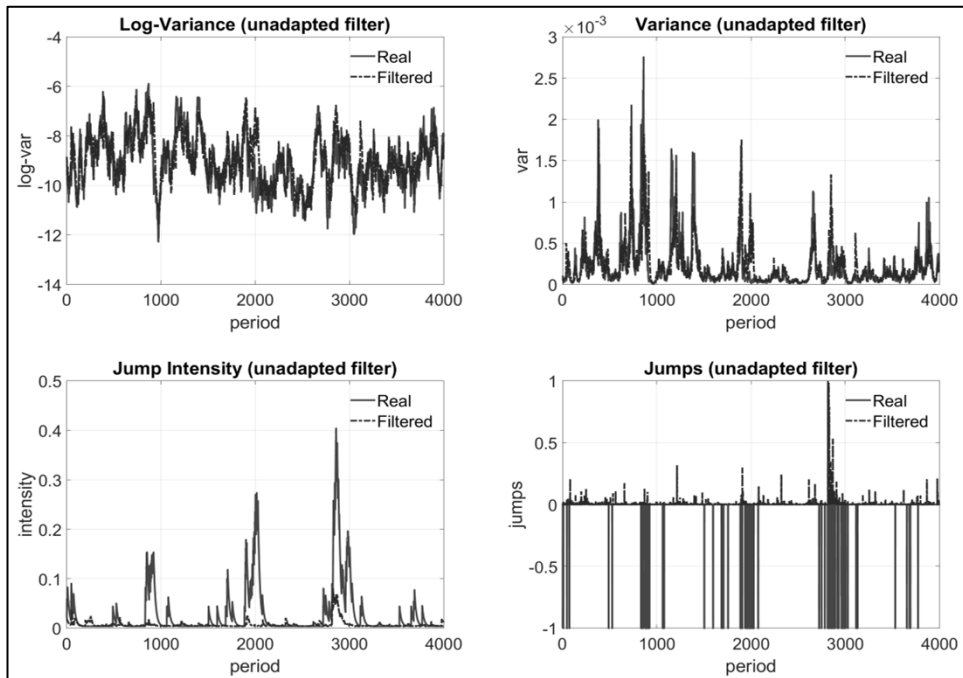
$$w_t^{(i)} = \frac{p(r_t | h_t^i, J_t^i, Q_t^i) p(J_t^i) (\lambda_t^i)^{Q_t^i} (1 - \lambda_t^i)^{1 - Q_t^i}}{p(J_t^i | h_t^i, r_t, Q_t^i) (\lambda_t^{i*})^{Q_t^i} (1 - \lambda_t^{i*})^{1 - Q_t^i}} \tilde{w}_{t-1}^{(i)} \quad (61)$$

8 Simulation Results

Time series of returns were simulated according to the described SVJD model with parameters given as: $\mu = \frac{0.05}{252}$, $\mu_J = -0.01$, $\sigma_J = 0.04$, $V_{LT} = e^{\frac{\alpha}{1-\beta}} = 0.01^2$, $\beta = 0.98$, $\gamma = 0.2$, $\lambda_{LT} = \frac{\alpha_J}{1-\beta_J-\gamma_J} = 0.02$, $\beta_J = 0.95$, and $\gamma_J = 0.04$.

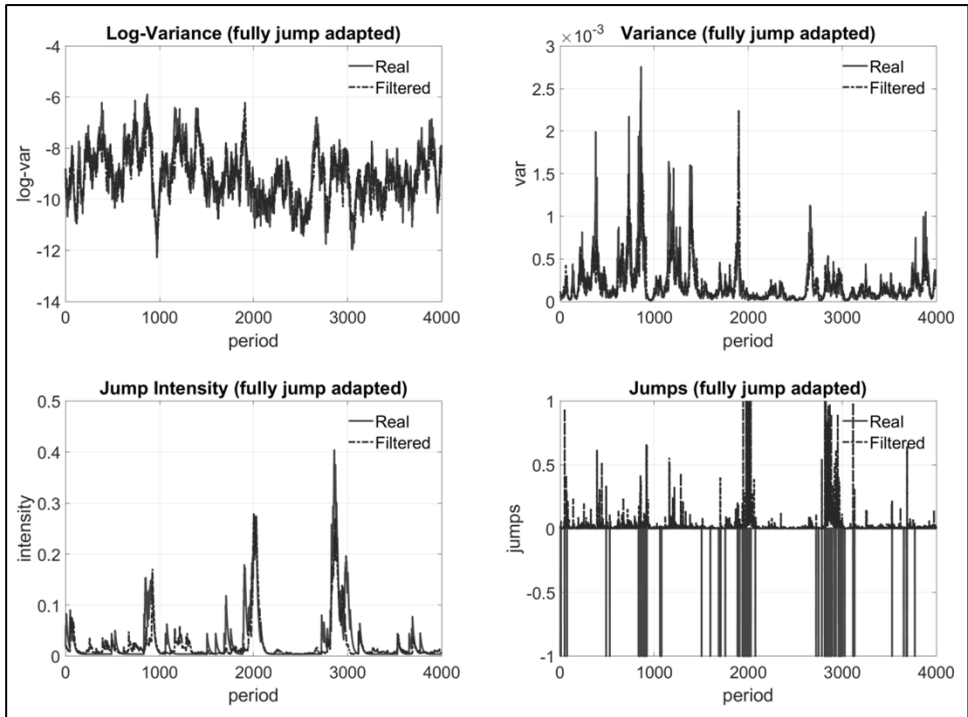
For illustration purposes, we first show the performance of a particle filter with only 100 particles (and a 50 particle re-sampling threshold) on a simulated time series with 4,000 observations, for the un-adapted filter, and for the filter with fully adapted jumps (i.e. the filter adapted to jump sizes and occurrences, described in Section 7).

Fig. 1 Simulated vs. filtered time series for the un-adapted particle filter



Source: Authorial computation.

Fig. 2 Simulated vs. filtered time series for the jump-adapted particle filter



Source: Authorial computation.

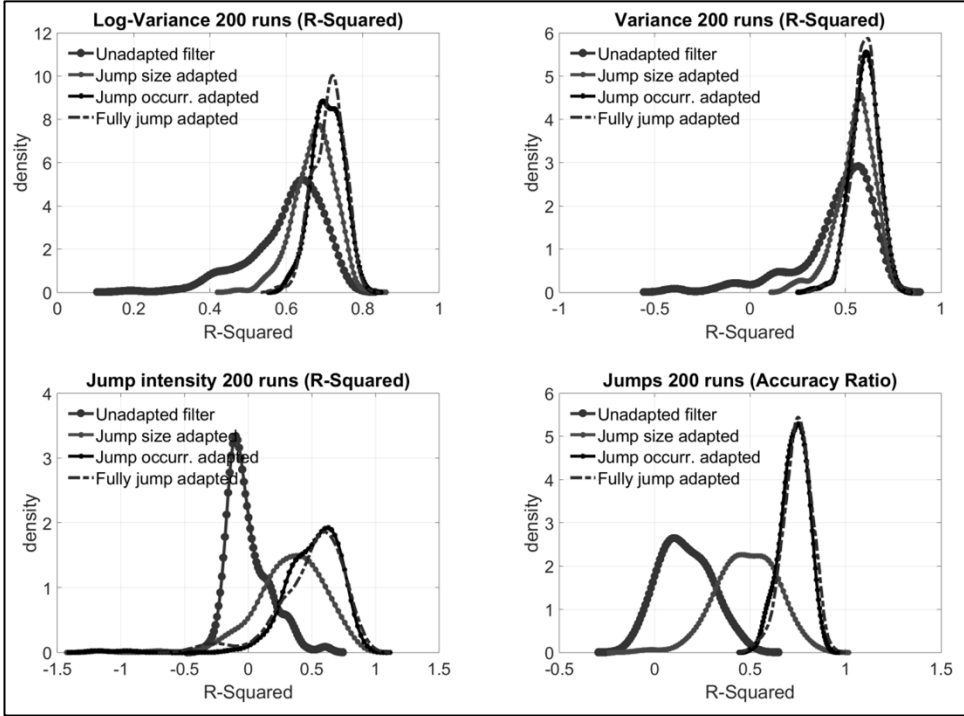
Figure 1 shows filtered latent state time series for the stochastic log-variances, variances, jump intensities and jump occurrences for the un-adapted particle filter, while Figure 2 shows the same filtered time series for the jump-adapted filter.

While we cannot see any pronounced differences in the log-variance and variance time series, we can see that the un-adapted particle filter estimated far less jumps, and that it struggled to capture the increases in jump intensity, caused by the jump self-excitation around the 2,000th and the 3,000th period of simulation. The adapted filter, on the other hand, managed to estimate larger number of jumps and it was also able to better capture the evolution of the jump intensity.

To compare the results more quantitatively, the simulation was run 200 times, and the four proposed filters (un-adapted, jump-size adapted, jump-occurrence adapted, and fully jump-adapted) were applied to each of the simulations. R-Squared (R2) was computed to assess how well do the filtered estimates fit the log-variance, variance and jump-intensity, and the Accuracy Ratio (AR) was used to evaluate how well do they manage to filter the simulated jumps. Figure 3 shows

the distributions of these quantities over the 200 simulations for all of the proposed filters.

Fig. 3 Distribution of the R2 and AR achieved in 200 simulation runs



Source: Authorial computation.

We can see that the un-adapted particle filter achieved by far the worst results, followed by the jump-size adapted filter and the jump-occurrence adapted filter, while the best performance was achieved by the fully jump-adapted particle filter.

The mean values of the R-Squared (R2) and the Accuracy Ratio (AR) achieved over the 200 runs of the simulation test are summarised in Table 1.

Tab. 1 Average R2 and AR of the filtered estimates in 200 simulation runs

| Particle Filter | log-variance R2 | variance R2 | jump intensity R2 | jump occurrence AR |
|----------------------|-----------------|-------------|-------------------|--------------------|
| Un-Adapted filter | 0.604 | 0.456 | -0.002 | 0.160 |
| Jump-size adapted | 0.673 | 0.553 | 0.326 | 0.484 |
| Jump-occurr. adapted | 0.708 | 0.599 | 0.501 | 0.732 |
| Fully jump-adapted | 0.711 | 0.601 | 0.490 | 0.747 |

Source: Authorial computation.

The numerical results in Table 1 clearly show the fully jump-adapted and the jump-occurrence adapted particle filters to be the best, while the un-adapted particle filter achieved the worst results, especially with regards to the filtering of the jumps.

9 Conclusion

Particle filter algorithms for the filtering of latent state time series of SVJD models were explained and several methods of how to adapt these filters to the jump occurrences and jump sizes were derived. The results of the simulation study show that the proposed jump-adapted particle filters significantly outperform the un-adapted particle filter in their ability to filter the latent volatility and jumps. The filter adapted to both the jump occurrences and jump sizes achieved on average the best results, followed narrowly by the filter adapted only to the jump occurrences. The performance of the filter adapted only to the jump sizes was lower, but still much better than for the un-adapted particle filter. In future studies it would be interesting to test particle filters approximately adapted to the stochastic variances as well, and to apply the methodology to more complex SVJD models, including jumps in volatility, correlation between volatility and returns, multiple volatility components, or other empirically observed effects of financial time series.

References

- Andersen, T. G., Bollerslev, T., Christoffersen, P. F., Diebold, F. X., 2005. Volatility Forecasting. National Bureau of Economic Research, NBER Working Paper No. 11188. DOI: 10.3386/w11188.
- Bates, D. S., 1996. Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options. *Review of Financial Studies* 9, 69-107. DOI: 10.1093/rfs/9.1.69.
- Eraker, B., Johannes, M., Polson, N., 2003. The Impact of Jumping Equity Index Volatility and Returns. *The Journal of Finance* 58, 1269-1300. DOI: 10.1111/1540-6261.00566.
- Fičura, M., Witzany, J., 2016. Estimating Stochastic Volatility and Jumps Using High-Frequency Data and Bayesian Methods. *Czech Journal of Economics and Finance* 66, 279-301. DOI: 10.2139/ssrn.2551807.
- Fulop, A., Li, J., 2013. Efficient Learning via Simulation: A Marginalized Resample-Move Approach. *Journal of Econometrics* 176, 146-161. DOI: 10.1016/j.jeconom.2013.05.002.
- Fulop, A., Li, J., Yu, J., 2015. Self-Exciting Jumps, Learning and Asset Pricing Implications. *Review of Financial Studies* 28, 876-912. DOI: 10.1093/rfs/hhu078.

Gordon, N. J., Salmond, D. J., Smith, A. F. M., 1993. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. IEE Proceedings F on Radar and Signal Processing 140, 107-113. DOI: 10.1049/ip-f-2.1993.0015.

Jacquier, E., Johannes, M., Polson, N., 2007. MCMC Maximum Likelihood for Latent State Models. Journal of Econometrics 137, 615-640. DOI: 10.1016/j.jeconom.2005.11.017.

Jacquier, E., Polson, N., Rossi, P., 1994. Bayesian Analysis of Stochastic Volatility Models. Journal of Business & Economic Statistics 12, 371-389. DOI: 10.1080/07350015.1994.10524553

Kim, S., Shephard, N., Chib, S., 1998. Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models. Review of Economic Studies 65, 361-393. DOI: 10.1111/1467-937x.00050.

Pitt, M. K., Shephard, N., 1999. Filtering via Simulation: Auxiliary Particle Filters. Journal of the American Statistical Association 94, 590-599. DOI: 10.1080/01621459.1999.10474153.

Shephard, N., 2005. Stochastic Volatility: Selected Readings (Advanced Texts in Econometrics). Oxford University Press, Oxford.

Witzany, J., 2013. Estimating Correlated Jumps and Stochastic Volatilities. Prague Economic Papers 22, 251-283. DOI: 10.18267/j.pep.451.