

# Particle Filter-Based On-Line Estimation of Spot Volatility with Nonlinear Market Microstructure Noise Models

Rainer Dahlhaus and Jan C. Neddermeyer

University of Heidelberg

**Summary.** A new technique for the on-line estimation of spot volatility for high-frequency data is developed. The algorithm works directly on the transaction data and updates the volatility estimate immediately after the occurrence of a new transaction. We make a clear distinction between volatility per time unit and volatility per transaction and provide estimators for both. A new nonlinear market microstructure noise model is proposed that reproduces the major stylized facts of high-frequency data. A computationally efficient particle filter is used that allows for the approximation of the unknown efficient prices and, in combination with a recursive EM algorithm, for the estimation of the volatility curves. In addition, the estimators are improved by an on-line bias correction. We neither assume that the transaction times are equidistant nor do we use interpolated prices.

**Keywords.** Nonlinear state-space model; microstructure noise; sequential EM algorithm; sequential Monte Carlo; tick-by-tick data; transaction time.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>A New Nonlinear Market Microstructure Noise Model</b>	<b>3</b>
<b>3</b>	<b>On-Line Estimation of Spot Volatility Based on a Particle Filter and Sequential EM-Type Algorithms</b>	<b>7</b>
3.1	A Nonlinear State-Space Model . . . . .	7
3.2	An Efficient Particle Filter . . . . .	8
3.3	A Sequential EM-Type Algorithm . . . . .	10
3.4	Summary . . . . .	13
<b>4</b>	<b>From Transaction Time to Clock Time</b>	<b>14</b>
4.1	Clock Time Spot Volatility Estimation . . . . .	14
4.2	An Alternative Estimator for Clock Time Spot Volatility . . . . .	15
<b>5</b>	<b>Fine-Tuning of the Volatility Estimator in the Time-Varying Case</b>	<b>16</b>
5.1	Adaptive Step Size Selection . . . . .	17
5.2	On-line Bias Correction and Mean Squared Error Minimization . . . . .	18
<b>6</b>	<b>Implementation Overview in the Time-Varying Case</b>	<b>22</b>
<b>7</b>	<b>Simulations and Applications</b>	<b>23</b>
7.1	Results for Simulated Data . . . . .	23
7.2	Results for Real Data . . . . .	26
<b>8</b>	<b>Concluding Remarks</b>	<b>30</b>
<b>9</b>	<b>Appendix</b>	<b>33</b>

## 1 Introduction

For high-frequency data occurring on a millisecond basis the on-line estimation of spot volatility is a challenging task because of the presence of market microstructure noise. In this article, the efficient log-price process of a security is treated as a latent state in a nonlinear state-space model. The relation between the efficient log-prices and the transaction prices is described by a new nonlinear market microstructure noise model. A computationally efficient particle filter is developed which allows the estimation of the filtering distributions of the efficient log-prices given the observed transaction prices. Based on the filtering distributions the time-varying volatility is estimated using a new sequential Expectation-Maximization (EM) algorithm. Our procedure works on-line and updates the volatility estimate immediately when a new transaction comes in. The method is suitable for real-time applications because of its computational efficiency.

Until recently, the main focus in the literature has been on the estimation of the integrated volatility. This task has been studied extensively under various assumptions on the market microstructure noise (Zhou 1996; Zhang et al. 2005; Andersen et al. 2006; Bandi and Russell 2006, 2008; Hansen and Lunde 2006; Barndorff-Nielsen et al. 2008; Kalnina and Linton 2008; Christensen et al. 2009; Jacod et al. 2009; Podolskij and Vetter 2009). Some authors suggested that estimates of the spot volatility can be obtained through localized versions of estimators for the integrated volatility (Harris 1990; Foster and Nelson 1996; Zeng 2003; Fan and Wang 2008; Bos et al. 2009; Kristensen 2009) or by (Fourier-) series methods (Munk and Schmidt-Hieber

2009). In contrast to these existing methods which are essentially off-line procedures, our approach allows on-line estimation.

In this article, transaction data are treated as noisy observations of a latent efficient log-price process  $X_t$ . We assume that transaction prices  $Y_{t_j}$  are observed at times  $t_1 < t_2 < \dots < t_T$ . The evolution of the efficient log-price process is modeled by a random walk in transaction time with possibly time-varying volatility  $\sigma_{t_j}$ , that is

$$X_{t_j} = X_{t_{j-1}} + Z_{t_j} \quad (1)$$

with  $Z_{t_j} \sim \mathcal{N}(0, \sigma_{t_j}^2)$ , or alternatively by a diffusion model in clock time – see Section 4. Drift terms are ignored because their effect is of lower order with high-frequency data.

We make a clear distinction between volatility per time unit and volatility per transaction and provide estimators for both. We start with a model in transaction time instead of clock time leading to an estimator of the spot volatility per transaction. In Section 4, a transformation from transaction time volatility to clock time volatility is given leading to a subsequent estimator of the volatility per time unit. In addition, we give a direct clock time estimator. In our opinion a model in transaction time has at least two advantages: First, the distribution of asset log-returns in a transaction time model can be modeled in most situations quite well by a Gaussian distribution, and second, volatility in transaction time is more constant than volatility in clock time making the algorithm more stable (Ané and Geman 2000; Plerou et al. 2001; Gabaix et al. 2003 – see also the discussion in sections 4, 7, and 8).

The relation between the efficient (log-)prices and the observed transaction prices is described through a general nonlinear market microstructure noise model which is completely different from the models considered so far. It depends on the (observed or unobserved) order book or market maker quotes and it can be expressed through a nonlinear equation

$$Y_{t_j} = g_{t_j}(\exp[X_{t_j}]) = g_{t_j; Y_{t_{1:j-1}}}(\exp[X_{t_j}]), \quad (2)$$

where the function  $g_{t_j}$  may also depend on past observations  $Y_{t_{1:j-1}} := \{Y_{t_1}, \dots, Y_{t_{j-1}}\}$  (see case 3 in Section 2). The function  $g_{t_j}$  is time-inhomogeneous and it can be interpreted as a generalized rounding function. The details of this model along with its economic motivation are given in Section 2.

The state equation (1) and the observation equation (2) form a nonlinear state-space model. The volatility is considered as a parameter of this state-space model. The estimation is done through a particle filter and a new sequential EM-type algorithm. Very roughly speaking our volatility estimator can be viewed as a localized realized volatility estimator based upon the particles of the particle filter. In detail the situation is however more complicated because we need a back and forth between particle filter and volatility estimator to obtain a decent on-line estimator. Bias corrections and an adaptive parameter choice complicate the situation even further.

We mention that our methods are not restricted to the above model but can also be applied

with other microstructure noise models. Contrary to several other papers we do not assume that the transaction times are equidistant nor do we use interpolated prices.

The article is organized as follows. Section 2 describes the nonlinear market microstructure noise model. In Section 3, a particle filter and a sequential EM-type algorithm are proposed for on-line estimation of spot volatility. In Section 4, estimation of spot volatility in clock time is sketched and a transformation from transaction time volatility to clock time volatility is given including the corresponding estimates. Methods for adaptive bias correction and step size selection are proposed in Section 5. A description of the implementation of our algorithm is given in Section 6. Finally, simulation results and an application to real data are presented in Section 7 followed by some conclusions in Section 8.

## 2 A New Nonlinear Market Microstructure Noise Model

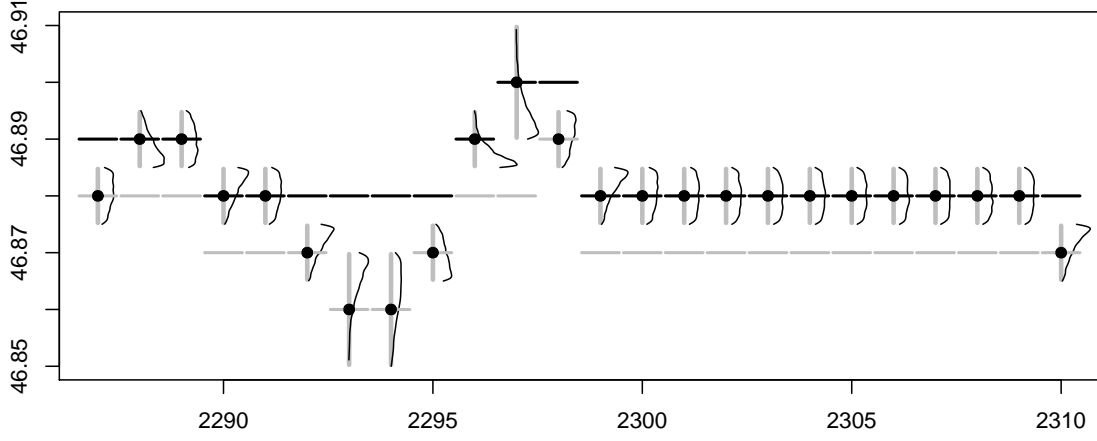
In most existing market microstructure models the efficient log-price is assumed to be corrupted by additive stationary noise (Aït-Sahalia et al. 2005; Zhang et al. 2005; Bandi and Russell 2006; Hansen and Lunde 2006; Barndorff-Nielsen et al. 2008). The noise variables are typically independent of the efficient log-price process. The major weakness of these models is the fact that they cannot reproduce the discreteness of the transaction prices. More adequate models which incorporate rounding noise have also been considered (Ball 1988; Large 2007; Li and Mykland 2007; Robert and Rosenbaum 2008; Rosenbaum 2009). A popular model is based on additive noise followed by rounding according to the smallest tick size. A drawback of most existing models is the dependence on parameters and on distributional assumptions.

In this article, a general market microstructure noise model is proposed which differs significantly from existing models. We are convinced that it is more suitable to explain microstructure features of real data. The model is based on the following simple assumption on the filtering distribution  $p(\exp[x_{t_j}]|y_{t_1}, \dots, y_{t_j})$  of the unknown efficient price  $\exp[X_{t_j}]$  given the observed transaction prices  $Y_{t_1} = y_{t_1}, \dots, Y_{t_j} = y_{t_j}$ .

**Model assumption 1:** The support  $A_{t_j}$  of the filtering distribution  $p(\exp[x_{t_j}]|y_{t_1}, \dots, y_{t_j})$  is bounded and known.

It follows that the support of the filtering distribution of the efficient log-price  $p(x_{t_j}|y_{t_1}, \dots, y_{t_j})$  is given by  $\log A_{t_j}$ .

This assumption is rather weak because we make no assumption at all on the distribution of  $Y_{t_j}$ . The clue is that given the model of the efficient log-price process (1) this assumption already leads to the identifiability of the distribution  $p(x_{t_j}|y_{t_1}, \dots, y_{t_j})$  (see Proposition 1 below). It is shown later that this distribution can be approximated through a particle filter. A real data example is given in Figure 1. It shows the supports  $A_{t_j}$  (gray vertical lines) and kernel density estimates of the filtering distributions of the efficient prices (black lines) which are computed based on the output of the particle filter. In this example, market maker quotes are available (see



**Figure 1:** A real data example of estimated filtering distributions based on our market microstructure noise model for the case when market maker quotes are available in addition to the transaction data. The details are provided in Section 7.2. The plot shows some transaction prices (circles) along with kernel density estimates of the filtering distributions of the efficient prices (black lines) based on the particles produced by our particle filter. The gray vertical lines indicate the assumed support of the filtering distributions. The bid and ask market maker quotes are displayed by gray and black horizontal lines, respectively. The x-axis shows transaction time.

case 2 below) which are indicated by gray and black horizontal lines. The details of this example are provided in Section 7.2.

The above model assumption is, for instance, fulfilled in the following three cases: In cases 1 and 2, limit order book data and market maker quotes are available, respectively, in addition to the transaction data leading to the support  $A_{t_j}$ . In case 3, only transaction data are available and a method to construct the  $A_{t_j}$  is suggested.

**Case 1:** (order book data)

Let's assume that at each transaction time  $t_j$  the exchange provides a limit order book with bid and ask levels given by  $\alpha_{t_j}^k$  and  $\beta_{t_j}^k$ ,  $k = 1, 2, \dots, K$ , respectively. The order book levels satisfy  $\alpha_{t_j}^K < \dots < \alpha_{t_j}^2 < \alpha_{t_j}^1 < \beta_{t_j}^1 < \beta_{t_j}^2 < \dots < \beta_{t_j}^K$  and we denote

$$\mathcal{M}_{t_j} = \{\alpha_{t_j}^K, \dots, \alpha_{t_j}^2, \alpha_{t_j}^1, \beta_{t_j}^1, \beta_{t_j}^2, \dots, \beta_{t_j}^K\}.$$

$\mathcal{M}_{t_j}$  represents the state of the order book immediately before the transaction at time  $t_j$  occurs. Clearly,  $y_{t_j} \in \mathcal{M}_{t_j}$ . The support of the filtering distribution at time  $t_j$  is defined through

$$A_{t_j} = \{x \in \mathbb{R} : \operatorname{argmin}_{\gamma \in \mathcal{M}_{t_j}} |x - \gamma| = y_{t_j}\}.$$

Thus, the transaction price at time  $t_j$  is that price in the set  $\mathcal{M}_{t_j}$  with the smallest Euclidean distance to the efficient price. Note, that  $A_{t_j}$  is simply an interval of the real line. The economic intuition behind this model is that the efficient price at time  $t_j$  should be closer to the observed price  $y_{t_j}$  than to any other order book level. Of course, this cannot be guaranteed. However, it seems to be more realistic assumption than many other microstructure noise models leading at the same time to quite strong results.

An example of this market microstructure model is visualized in Figure 2. The supports of the filtering distributions are indicated by thick vertical lines. Observe that sometimes the bid-ask spread widens the support of the filtering distribution.

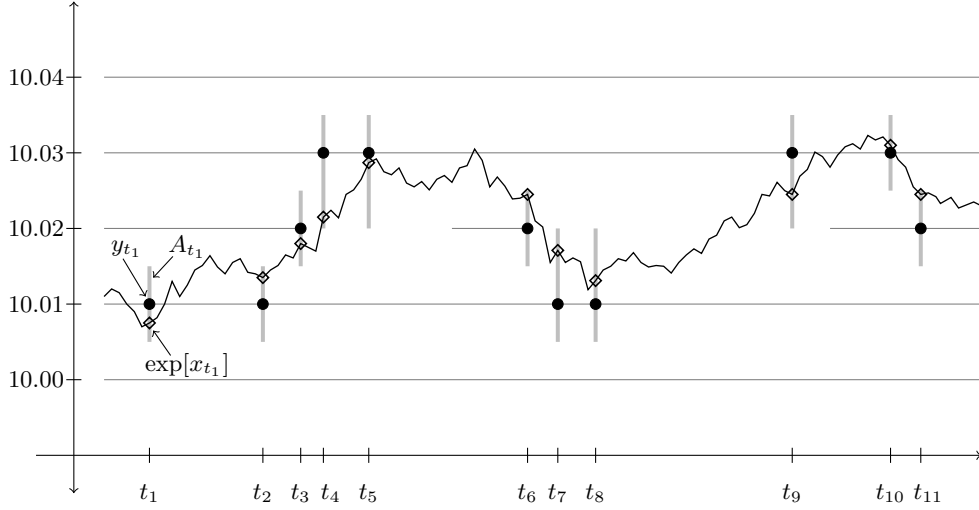


Figure 2: An example of our market microstructure noise model for the case when order book data are available. The figure shows the transaction prices (circles), the (in practice unknown) efficient prices in transaction time (diamonds), the latent efficient price process in clock time (black line), the order book levels (gray horizontal lines), and the supports of the filtering distributions of the efficient prices (gray vertical lines).

### Case 2: (market maker quotes)

In the case where market maker quotes are available (instead of order book data), we only have a single bid and a single ask level  $\alpha_{t_j}$  and  $\beta_{t_j}$ , respectively, which satisfy  $\alpha_{t_j} < \beta_{t_j}$ . That is,  $y_{t_j}$  is either equal to  $\alpha_{t_j}$  or equal to  $\beta_{t_j}$ . The supports  $A_{t_j}$  are then defined through

$$A_{t_j} = [y_{t_j} - \Delta_{t_j}, y_{t_j} + \Delta_{t_j}),$$

where  $\Delta_{t_j} = 0.5(\beta_{t_j} - \alpha_{t_j})$ . The economic intuition given in case 1 applies similarly.

### Case 3: (transaction data only)

For the case where no order book data or market maker quotes are available we now suggest a method for defining the supports of the filtering distributions solely based on the observed transaction prices. Conditional on  $y_{t_1}, \dots, y_{t_j}$ , we set

$$A_{t_j} = [y_{t_j} - \Delta_{t_j}, y_{t_j} + \Delta_{t_j}),$$

where

$$\Delta_{t_j} = \begin{cases} 0.5|y_{t_j} - y_{t_{j-1}}| & \text{if } y_{t_j} \neq y_{t_{j-1}}, \\ \Delta_{t_{j-1}} & \text{else.} \end{cases}$$

Note that  $\Delta_{t_j}$  can be seen as an estimate of half the bid-ask spread at time  $t_j$ .

In practice, the intervals  $A_{t_j}$  will be similar for all three cases. Consequently, the estimation results will not differ much. It is mentioned that we do not need to explicitly specify the unknown nonlinear function  $g_{t_j}$  in the observation equation (2). The model assumption can be regarded as an assumption on the inverse mapping  $g_{t_j}^{-1}$ , namely  $g_{t_j}^{-1}(y_{t_j}) = \{x | g_{t_j}(x) = y_{t_j}\} = A_{t_j}$  (conditional on  $y_{t_1}, \dots, y_{t_{j-1}}$ ). That is, the observed price  $y_{t_j}$  determines the possible values of the associated efficient price.

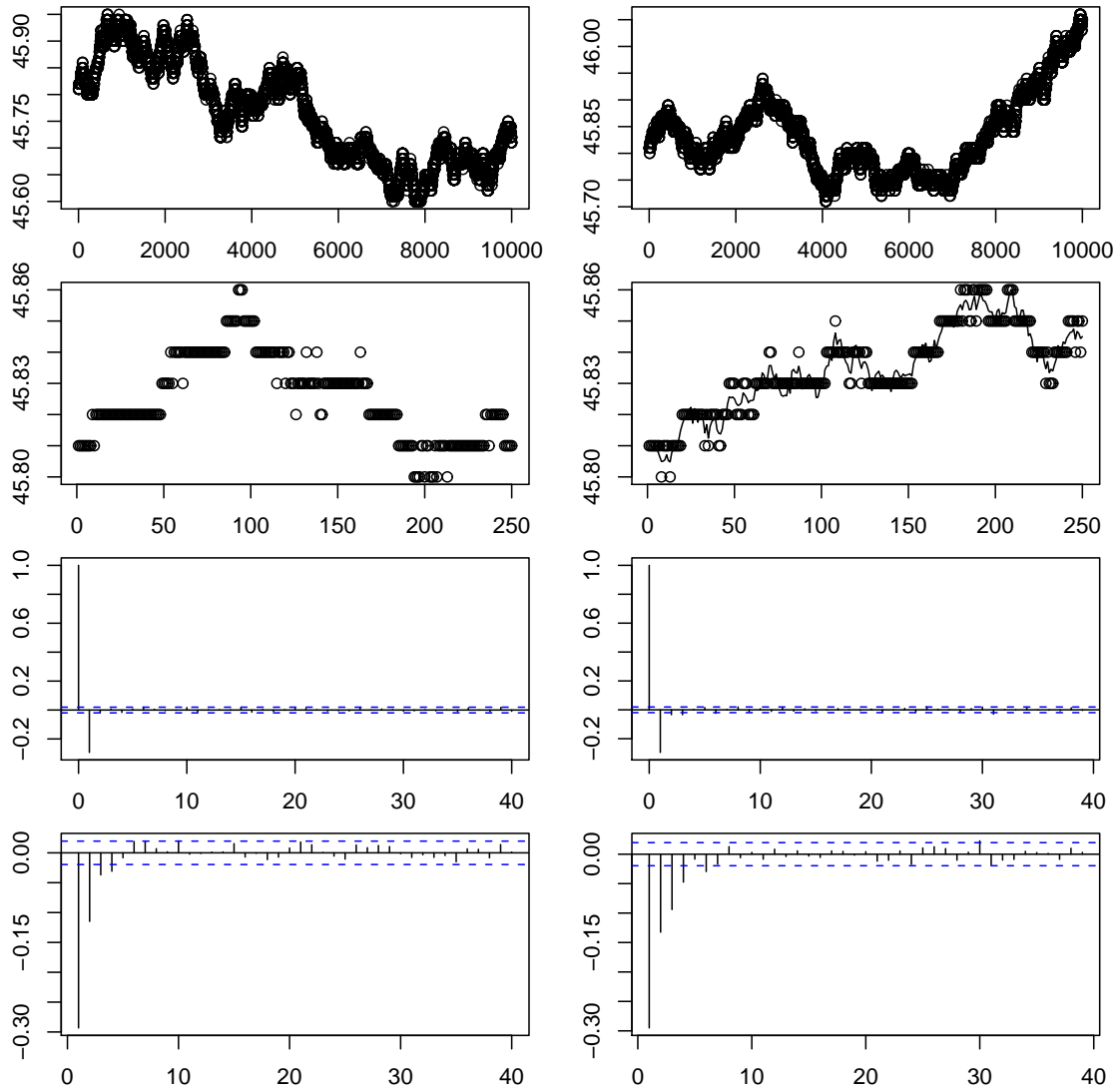


Figure 3: Comparison of real transaction data for Citigroup (left column) with simulated data from our market microstructure noise model (right column). The plots show (from top to bottom): 10,000 transaction prices; the first 250 transaction prices and the efficient price process of the simulated data; the autocorrelations and the partial autocorrelations of the returns of the transaction prices.

We strongly believe that our model better describes the real world market microstructure than most existing models. Data simulated from our model reproduce the major stylized facts of high-frequency data, such as price discreteness and a first-order negative autocorrelation of the returns. Therefore, it seems to be an adequate model. As an example, transaction data of Citigroup are compared with data simulated from our model (see Figure 3). The figure shows the simulated efficient prices and the observations. The observations are the efficient prices rounded to the nearest cent (that is  $\Delta_{t_j} \equiv 0.5$  cents). The efficient log-prices were generated according to (1) such that the observations have approximately the same volatility as the Citigroup data. We emphasize the large number of zero returns in both data. It is not surprising that the trajectories of the transaction processes look completely different. The important point, however, is that our market microstructure noise model automatically introduces autocorrelations and partial autocorrelations of the returns which are very similar to those of the real Citigroup data.

Our estimation method is not limited to this market microstructure noise model. In particular it can also be applied after a suitable modification of the particle filter to the additive microstructure noise models

$$Y_{t_j} = \text{round}(\exp[X_{t_j}] + U_{t_j}) \quad \text{or} \quad Y_{t_j} = \text{round}(\exp[X_{t_j} + U_{t_j}]) \quad (3)$$

where the  $U_{t_j}$  are for instance i.i.d. Gaussian distributed.

### 3 On-Line Estimation of Spot Volatility Based on a Particle Filter and Sequential EM-Type Algorithms

We now present on-line algorithms for the estimation of the spot volatility. Because all results also hold in the multivariate case with synchronous trading times we formulate this section for multivariate security prices. We are aware of the fact that the main challenge in the multivariate case are non-synchronous trading times. The presented results are, however, the basis for future work on non-synchronous trading.

We therefore consider in this section the estimation of the covariance matrix  $\Sigma_{t_j}$  which gives the volatilities of the individual efficient log-price processes  $X_{t,s}$ ,  $s = 1, \dots, S$ , as well as their cross-volatilities. The algorithms for the spot volatility are obtained by setting  $\Sigma_{t_j} = \sigma_{t_j}^2$ .

#### 3.1 A Nonlinear State-Space Model

The multivariate version of the nonlinear state-space model (2) and (1) is given by

$$\mathbf{Y}_{t_j} = g_{t_j}(\exp[\mathbf{X}_{t_j}]), \quad (4)$$

$$\mathbf{X}_{t_j} = \mathbf{X}_{t_{j-1}} + \mathbf{Z}_{t_j}, \quad (5)$$

where  $\mathbf{X}_t = (X_{t,1}, \dots, X_{t,S})^T$ ,  $g_{t_j}(\exp[\mathbf{X}_{t_j}]) = (g_{t_j^{(1)}}(\exp[X_{t_j,1}]), \dots, g_{t_j^{(S)}}(\exp[X_{t_j,S}]))^T$  with the  $g_{t_j^{(s)}}$  possibly depending on  $Y_{t_{1:j-1}}$ , and  $\mathbf{Z}_{t_j} \sim \mathcal{N}(\mathbf{0}, \Sigma_{t_j})$ . The set  $\mathbf{A}_{t_j}$  from Model assumption 1 usually is of the form  $\mathbf{A}_{t_j} = A_{t_j,1} \times \dots \times A_{t_j,S}$  with the  $A_{t_j,s}$  being intervals (although this is not used). For simplicity we assume as an initial condition that given  $Y_{t_{1,s}}$  the efficient prices  $\exp[X_{t_{1,s}}]$  are uniformly distributed on  $A_{t_{1,s}}$ .

**Model assumption 2:**  $\Sigma_{t_j}$  is assumed to be either constant or slowly varying in time, that is we assume some smoothness for  $\Sigma_{t_j}$ .

The smoothness assumption does not need to be specified any further because we do not use it formally. However, without this assumption the estimation procedure developed in Section 3.3 would not make sense. A detailed specification of this assumption would become necessary if we tried to prove consistency (see Section 8).

We remark that (4) and (5) constitute a slightly generalized state-space model because the observations  $\mathbf{Y}_{t_j}$  are not conditional independent of  $\mathbf{Y}_{t_{1:j-1}}$  given  $\mathbf{X}_{t_j}$  as in standard state-space models. This dependency on past observations is induced by our market microstructure noise



model (see case 3 in Section 2). In the following section a particle filter is derived which can cope with this setting.

Our objective is the estimation of the covariance matrix  $\Sigma_{t_j}$  based on the observed prices  $\mathbf{Y}_{t_{1:j}} = \mathbf{y}_{t_{1:j}}$ . Because of the nonlinear market microstructure noise this is difficult. It is well known that crude estimators that ignore the noise lead to severely biased estimates (see, for instance, Voev and Lunde 2007). The idea of our estimation procedure is to approximate the conditional distribution of the efficient log-prices  $\mathbf{X}_{t_j}$  given all observed transaction prices  $\mathbf{y}_{t_{1:j}}$  up to time  $t_j$  by an efficient particle filter. Based on this approximation a localized EM-type algorithm is used to construct an estimator of  $\Sigma_{t_j}$ .

### 3.2 An Efficient Particle Filter

Particle filters are sequential Monte Carlo methods (Doucet et al. 2001) that approximate the posterior distributions  $p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}})$  with clouds of particles  $\{\mathbf{x}_{t_{1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$ . A particle consists of a sample  $\mathbf{x}_{t_{1:j}}^i$  and an associated weight  $\omega_{t_j}^i$ . The particle approximation of the target distribution is given by

$$p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}}) \approx \sum_{i=1}^N \omega_{t_j}^i \delta_{\mathbf{x}_{t_{1:j}}^i}(\mathbf{x}_{t_{1:j}}),$$

with  $\delta$  being the Dirac delta function. A particle filter generates particles sequentially in time making use of the relation

$$p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}}) = \frac{p(\mathbf{y}_{t_j}, \mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j-1}})}{p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}})} = \frac{p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}})}{p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}})} p(\mathbf{x}_{t_{1:j-1}} | \mathbf{y}_{t_{1:j-1}}) \quad (6)$$

and a general sampling technique known as importance sampling. Importance sampling is necessary because direct sampling from (6) is not feasible. In standard state-space models  $p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})$  further simplifies to  $p(\mathbf{y}_{t_j} | \mathbf{x}_{t_j})$ . As a result of the violated conditional independence property mentioned earlier, this is not the case for our model.

In each iteration of the particle filter samples are drawn from an importance sampling distribution called proposal. Subsequently, the samples are weighted such that they approximate the target distribution. The choice of the proposal is crucial for the efficiency of the filter. In our framework it is possible to sample from the proposal  $p(\mathbf{x}_{t_j} | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}})$  which is the optimal proposal in the sense that it minimizes the variance of the importance sampling weights (Doucet et al. 2000). The following algorithm is known as sequential importance sampling (Gordon et al. 1993): Assume that weighted particles  $\{\mathbf{x}_{t_{1:j-1}}^i, \omega_{t_{j-1}}^i\}_{i=1}^N$  approximating  $p(\mathbf{x}_{t_{1:j-1}} | \mathbf{y}_{t_{1:j-1}})$  are given; then

- For  $i = 1, \dots, N$ :
  - Sample  $\mathbf{x}_{t_j}^i \sim p(\mathbf{x}_{t_j} | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}^i)$ .
  - Compute importance weights

$$\tilde{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \frac{p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}^i) p(\mathbf{x}_{t_j}^i | \mathbf{x}_{t_{j-1}}^i)}{p(\mathbf{x}_{t_j}^i | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}^i)} = \omega_{t_{j-1}}^i p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_{j-1}}^i).$$

- For  $i = 1, \dots, N$ :
  - Normalize importance weights  $\omega_{t_j}^i = \check{\omega}_{t_j}^i / (\sum_{k=1}^N \check{\omega}_{t_j}^k)$ .
- Obtain particles  $\{\mathbf{x}_{t_{1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$  which approximate  $p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}})$ .

It is well-known that this algorithm suffers from weight degeneracy which means that after some iterations only few particles will have significant weight. This issue can be resolved by introducing a resampling step that maps the particle system  $\{\mathbf{x}_{t_{1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$  onto an equally weighted particle system  $\{\mathbf{x}_{t_{1:j}}^i, 1/N\}_{i=1}^N$ . Because resampling is time-consuming, it is carried out only if the effective sample size

$$\text{ESS}(\{\omega_{t_j}^i\}_{i=1}^N) = \frac{1}{\sum_{i=1}^N (\omega_{t_j}^i)^2}$$

is below some threshold (Kong et al. 1994). Different resampling schemes are discussed in Douc et al. (2005).

To apply the particle filter to the state-space model given by (4) and (5) it is necessary to specify the optimal proposal and the computation of the importance weights. The following result shows that both take a very simple form. Furthermore, it gives the uniqueness of the joint filtering distribution  $p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}})$ . This implies that in our microstructure noise model the knowledge of the support  $\mathbf{A}_{t_j}$  of  $p(\exp[\mathbf{x}_{t_j}] | \mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_j})$  already is sufficient for the identifiability of the efficient (log-)price distribution conditional on the observations.

**Proposition 1.** *The joint filtering distribution  $p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}})$  is uniquely determined by the supports  $\log \mathbf{A}_{t_k}$  of the filtering distributions  $p(\mathbf{x}_{t_k} | \mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_k})$ ,  $k = 1, \dots, j$ . The optimal proposal is a truncated multivariate normal distribution given by*

$$p(\mathbf{x}_{t_j} | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}) \propto \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}; \Sigma_{t_j}) \Big|_{\log \mathbf{A}_{t_j}}$$

with  $\log \mathbf{A}_{t_j} = \log A_{t_j,1} \times \dots \times \log A_{t_j,S}$  and the importance weights can be computed through

$$\check{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \int_{\log \mathbf{A}_{t_j}} \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i; \Sigma_{t_j}) d\mathbf{x}_{t_j}. \quad (7)$$

The proof can be found in the appendix.

**Remark:** For the market microstructure noise models (3) the optimal proposal cannot be computed that easily. For this case we suggest the standard proposal  $p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}) = \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}; \Sigma_{t_j})$  with importance weights

$$\check{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \int_{\log \mathbf{A}_{t_j}} \mathcal{N}(\mathbf{y} | \mathbf{x}_{t_j}^i; \sigma_U^2) d\mathbf{y} \quad \text{or} \quad \check{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \int_{\mathbf{A}_{t_j}} \mathcal{N}(\mathbf{y} | \exp[\mathbf{x}_{t_j}^i]; \sigma_U^2) d\mathbf{y},$$

respectively. This may necessitate a larger sample size  $N$ .

### 3.3 A Sequential EM-Type Algorithm

In this section, we discuss the estimation of  $\Sigma_{t_j}$  in the time-constant and time-varying case.

A stochastic EM algorithm can be used to obtain the maximum likelihood estimator in the time-constant case  $\Sigma_{t_j} = \Sigma$  (Dempster et al. 1977). The EM algorithm maximizes the likelihood  $p_{\Sigma}(\mathbf{y}_{t_{1:T}})$  by iteratively carrying out an E-step and an M-step. In the E-step, the expectation

$$\begin{aligned} \mathcal{Q}(\Sigma | \hat{\Sigma}^{(m)}) &= \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_{1:T}}, \mathbf{y}_{t_{1:T}}) | \mathbf{y}_{t_{1:T}}] \\ &= \sum_{j=1}^T \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{X}_{t_j}) | \mathbf{y}_{t_{1:T}}] + \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p(\mathbf{X}_{t_1}) | \mathbf{y}_{t_{1:T}}] \\ &\quad + \sum_{j=2}^T \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_j} | \mathbf{X}_{t_{j-1}}) | \mathbf{y}_{t_{1:T}}] \end{aligned} \quad (8)$$

needs to be approximated, where  $\hat{\Sigma}^{(m)}$  is the current estimator. Note, it is sufficient to consider the sum in (8) because the random variables  $\log p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{X}_{t_j})$  and  $p(\mathbf{X}_{t_1})$  do not depend on  $\Sigma$ . In the M-step, a new parameter estimate  $\hat{\Sigma}^{(m+1)}$  is obtained by maximizing  $\mathcal{Q}(\Sigma | \hat{\Sigma}^{(m)})$ .

If  $\Sigma_{t_j}$  is time-varying some regularization is needed. For example  $\hat{\Sigma}_{t_j}^{(m+1)}$  can be obtained by maximizing some localized version of (8), e.g.

$$\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}}^{(m)}) = \frac{1}{T} \sum_{k=j-T}^{j-2} \frac{1}{b} K\left(\frac{k}{bT}\right) \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}} | \mathbf{X}_{t_{j-k-1}}) | \mathbf{y}_{t_{1:T}}] \quad (9)$$

with a kernel  $K(\cdot)$  and a bandwidth  $b$ .

An approximation of  $\mathcal{Q}(\Sigma | \hat{\Sigma}^{(m)})$  and  $\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}}^{(m)})$  can be computed based on the smoothing particles

$$\{\mathbf{x}_{t_{1:T}}^i, \omega_{t_T}^i\}_{i=1}^N$$

from our particle filter or (with higher precision) from existing particle smoothing algorithms (Godsill et al. 2004; Neddermeyer 2010; Briers et al. 2010). The smoothing particles give the approximation

$$\begin{aligned} &\mathbf{E}_{\hat{\Sigma}_{t_{1:T}}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}} | \mathbf{X}_{t_{j-k-1}}) | \mathbf{y}_{t_{1:T}}] \\ &\approx \sum_{i=1}^N \omega_{t_T}^i \frac{1}{2} \left[ S \log 2\pi + \log |\Sigma| + \text{tr} \left\{ \Sigma^{-1} (\mathbf{x}_{t_{j-k}}^i - \mathbf{x}_{t_{j-k-1}}^i) (\mathbf{x}_{t_{j-k}}^i - \mathbf{x}_{t_{j-k-1}}^i)^T \right\} \right] \end{aligned} \quad (10)$$

which leads, with

$$\check{\Sigma}_{t_j}(\omega_{t_T}) := \sum_{i=1}^N \omega_{t_T}^i (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i) (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T, \quad (11)$$

to the maximizers

$$\hat{\Sigma}^{(m+1)} = \frac{1}{T-1} \sum_{j=2}^T \check{\Sigma}_{t_j}(\omega_{t_T}) \quad (12)$$

and

$$\hat{\Sigma}_{t_j}^{(m+1)} = \left[ \sum_k K\left(\frac{k}{bT}\right) \right]^{-1} \sum_k K\left(\frac{k}{bT}\right) \check{\Sigma}_{t_{j-k}}(\omega_{t_T}) \quad (13)$$

of (8) and (9), respectively (note that the particles and, therefore, also  $\check{\Sigma}$  depend on  $m$ .)

Instead of these estimates, one will prefer in most situations an on-line algorithm which updates the estimates when a new observation comes in. This requires on the one hand the use of filtering particles instead of smoothing particles and on the other hand an integration of the E-step into the algorithm.

We now develop such an algorithm step-by-step. Note that the recursion developed in 1) below is not an on-line algorithm. It is just discussed to demonstrate the relation of the on-line algorithms in (21) and (22) to the estimates (12) and (13), respectively. Note, in the following steps the notation  $\hat{\Sigma}_{t_j}$  is used for different estimates.

1) A “recursive” solution for the above situation (both for time-constant and time-varying  $\Sigma_{t_j}$ ) is

$$\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_1:T}) := \{1 - \lambda_j\} \mathcal{Q}_{t_{j-1}}(\Sigma|\hat{\Sigma}_{t_1:T}) + \lambda_j \mathbf{E}_{\hat{\Sigma}_{t_1:T}} [\log p_{\Sigma}(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_1:T}] \quad (14)$$

with  $\mathcal{Q}_{t_2}(\Sigma|\hat{\Sigma}_{t_1:T}) = \mathbf{E}_{\hat{\Sigma}_{t_1:T}} [\log p_{\Sigma}(\mathbf{X}_{t_2}|\mathbf{X}_{t_1})|\mathbf{y}_{t_1:T}]$  leading to

$$\begin{aligned} \mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_1:T}) &= \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \mathbf{E}_{\hat{\Sigma}_{t_1:T}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}}|\mathbf{X}_{t_{j-k-1}})|\mathbf{y}_{t_1:T}] \\ &\quad + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \mathbf{E}_{\hat{\Sigma}_{t_1:T}} [\log p_{\Sigma}(\mathbf{X}_{t_2}|\mathbf{X}_{t_1})|\mathbf{y}_{t_1:T}]. \end{aligned} \quad (15)$$

With the “constant parameter setting”  $\lambda_j := 1/(j - 1)$ , where  $\left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} = \frac{1}{j-1}$ , this gives the classical (quasi-) likelihood

$$\frac{1}{j-1} \sum_{k=0}^{j-2} \mathbf{E}_{\hat{\Sigma}_{t_1:T}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}}|\mathbf{X}_{t_{j-k-1}})|\mathbf{y}_{t_1:T}],$$

that is (8) for  $j = T$ . Furthermore, the maximizer of (15) is, with the smoother-approximation as in (10) and  $\check{\Sigma}_{t_j}(\omega_{t_T})$  as in (11), given by

$$\hat{\Sigma}_{t_j}^{(m+1)} = \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \check{\Sigma}_{t_{j-k}}(\omega_{t_T}) + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \check{\Sigma}_{t_2}(\omega_{t_T}). \quad (16)$$

This can be written as the recursion

$$\hat{\Sigma}_{t_j}^{(m+1)} = \{1 - \lambda_j\} \hat{\Sigma}_{t_{j-1}}^{(m+1)} + \lambda_j \check{\Sigma}_{t_j}(\omega_{t_T})$$

with  $\hat{\Sigma}_{t_2}^{(m+1)} = \check{\Sigma}_{t_2}(\omega_{t_T})$ . Again, we obtain with the “constant parameter setting”  $\lambda_j := 1/(j - 1)$  that  $\hat{\Sigma}_{t_j}^{(m+1)}$  coincides with the estimate in (12) for  $j = T$ .

2) On-line algorithms: The above algorithm is not an on-line algorithm because the conditional expectation in (14) depends on all observations. Therefore, we replace the conditioning set of variables  $\{\mathbf{y}_{t_1:T}\}$  by  $\{\mathbf{y}_{t_1:j}\}$  meaning that we pass from the smoothing distribution to the filtering distribution. More precisely,  $\mathbf{E}_{\hat{\Sigma}_{t_1:T}} [\log p_{\Sigma}(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_1:T}]$  is replaced by

$\mathbf{E}_{\hat{\Sigma}_{t_1:j-1}} [\log p_{\Sigma}(\mathbf{X}_{t_j} | \mathbf{X}_{t_{j-1}}) | \mathbf{y}_{t_1:j}]$  (we need at this point an estimate for  $\Sigma_{t_j}$  - see the comment at the end of this section) leading to the on-line algorithm

$$\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_1:j-1}) := \{1 - \lambda_j\} \mathcal{Q}_{t_{j-1}}(\Sigma | \hat{\Sigma}_{t_1:j-2}) + \lambda_j \mathbf{E}_{\hat{\Sigma}_{t_1:j-1}} [\log p_{\Sigma}(\mathbf{X}_{t_j} | \mathbf{X}_{t_{j-1}}) | \mathbf{y}_{t_1:j}] \quad (17)$$

with  $\mathcal{Q}_{t_2}(\Sigma | \hat{\Sigma}_{t_1}) = \mathbf{E}_{\hat{\Sigma}_{t_1}} [\log p_{\Sigma}(\mathbf{X}_{t_2} | \mathbf{X}_{t_1}) | \mathbf{y}_{t_1:2}]$ . (15) holds analogously and we now obtain analogous to (16) the estimate

$$\hat{\Sigma}_{t_j} = \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \check{\Sigma}_{t_{j-k}}(\omega_{t_{j-k}}) + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \check{\Sigma}_{t_2}(\omega_{t_2}) \quad (18)$$

now with

$$\check{\Sigma}_{t_j}(\omega_{t_j}) := \sum_{i=1}^N \omega_{t_j}^i (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i) (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T$$

based on the filtering particles  $\{\mathbf{x}_{t_{j-1}:j}^i, \omega_{t_j}^i\}_{i=1}^N$ . This estimate can be obtained from the on-line recursion

$$\hat{\Sigma}_{t_j} = \{1 - \lambda_j\} \hat{\Sigma}_{t_{j-1}} + \lambda_j \check{\Sigma}_{t_j}(\omega_{t_j}) \quad \text{with} \quad \hat{\Sigma}_{t_2} = \check{\Sigma}_{t_2}(\omega_{t_2}). \quad (19)$$

Observe that the estimated covariance matrix  $\hat{\Sigma}_{t_j}$  is positive semi-definite by construction.

The new parameter estimate  $\hat{\Sigma}_{t_j}$  is used afterwards to calculate the next filtering particles and their weights  $\{\mathbf{x}_{t_{j+1}}^i, \omega_{t_{j+1}}^i\}_{i=1}^N$  followed by the calculation of  $\hat{\Sigma}_{t_{j+1}}$  via another application of (19) etc. In contrast to the standard EM algorithm, our sequential variant therefore updates the covariance estimate (which in turn is used in the next step of the particle filter) in every time step. In the “new E-step”,  $\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_1:j-1})$  is approximated through

$$\begin{aligned} \hat{\mathcal{Q}}_{t_j}(\Sigma | \hat{\Sigma}_{t_1:j-1}) &= \{1 - \lambda_j\} \hat{\mathcal{Q}}_{t_{j-1}}(\Sigma | \hat{\Sigma}_{t_1:j-2}) \\ &\quad - \lambda_j \frac{1}{2} \sum_{i=1}^N \omega_{t_j}^i \left[ S \log 2\pi + \log |\Sigma| + \text{tr} \left\{ \Sigma^{-1} (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i) (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T \right\} \right] \end{aligned} \quad (20)$$

using the particles  $\{\mathbf{x}_{t_{j-1}:j}^i, \omega_{t_j}^i\}_{i=1}^N$  which are generated as described in Section 3.3. In the “new M-step”, the maximization of  $\hat{\mathcal{Q}}_{t_j}(\Sigma | \hat{\Sigma}_{t_1:j-1})$  gives the on-line estimator defined in (19).

Note that  $\check{\Sigma}_{t_j}(\omega_{t_j})$  is not an approximation of the conditional variance  $\text{Var}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}} | \mathbf{y}_{t_1:j})$  but an approximation of  $\mathbf{E}((\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^2 | \mathbf{y}_{t_1:j})$  (both are different because  $\mathbf{E}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}} | \mathbf{y}_{t_1:j}) \neq 0$ ). As a result of  $\mathbf{E}[\mathbf{E}((\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^2 | \mathbf{Y}_{t_1:j})] = \mathbf{E}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^2 = \text{Var}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})$ ,  $\hat{\Sigma}_{t_j}$  is a descent estimator of  $\Sigma_{t_j}$  apart from the bias problems described in Section 5.2.

3) Time-constant covariance matrices: If  $\Sigma_{t_j}$  is time-constant the first idea is to apply the algorithm (19) with the “constant parameter setting”  $\lambda_j = 1/(j-1)$ . However, the situation is different from the classical case in that the “old” estimate  $\hat{\Sigma}_{t_{j-1}}$  has in addition some bias due to the use of particles generated with an estimated covariance instead of the true one. Therefore we need to put less weight on the first term in (19). The situation has been carefully investigated for a similar algorithm in the i.i.d.-case by Cappé and Moulines (2009). Following their recommendation we use in our situation the on-line algorithm

$$\hat{\Sigma}_{t_j} = \{1 - (j-1)^{-\gamma}\} \hat{\Sigma}_{t_{j-1}} + (j-1)^{-\gamma} \check{\Sigma}_{t_j}(\omega_{t_j}) \quad (21)$$

with  $\gamma \in (\frac{1}{2}, 1)$ . Cappé and Moulines prove consistency and asymptotic normality of their estimate for weights  $\lambda_j := \lambda_0 j^{-\gamma}$  and  $\gamma \in (\frac{1}{2}, 1)$  and also for  $\gamma = 1$  under some restrictions on  $\lambda_0$  (Theorem 2). Furthermore, in their simulations it turned out that a value of  $\gamma = 0.6$  and  $\lambda_0 = 1$  has lead to good estimates. From our experience we prefer the choice  $\gamma = 0.9$  and  $\lambda_0 = 1$  (see Section 6). Even-Dar and Mansour (2003) obtained an optimal value of about 0.85 in a related estimation problem. We emphasize that the choice of  $\gamma$  needs more investigations - both theoretical and practical.

4) Time-varying covariance matrices: If  $\Sigma_{t_j}$  is time-varying it is necessary to put more weight on recent observations. In this case, the traditional solution is to use the algorithms (14), (17) and (19) with time-constant  $\lambda_j \equiv \lambda$  instead of a decaying  $\lambda_j$ . To achieve a better degree of adaptation our  $\lambda_j$  will still be time-varying (see Section 5.1) but with the intuition that the  $\lambda_j$  fluctuate around some fixed value of  $\lambda$ . That is we use in the time-varying case

$$\hat{\Sigma}_{t_j} = \{1 - \lambda_j\} \hat{\Sigma}_{t_{j-1}} + \lambda_j \check{\Sigma}_{t_j}(\omega_{t_j}) \quad \text{with} \quad \hat{\Sigma}_{t_2} = \check{\Sigma}_{t_2}(\omega_{t_2}). \quad (22)$$

The choice of the  $\lambda_j$  is discussed in Section 5.1. For a deeper understanding we stress the following heuristics: If  $\lambda_j \equiv \lambda$  and  $t_j = j \delta$  (e.g.  $\delta = \frac{1}{T}$ ) then we have with  $b := \frac{\delta}{\lambda}$  for  $\delta \rightarrow 0$

$$\left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} = (1 - \lambda)^k \lambda = \frac{\delta}{b} \left(1 - \frac{\delta}{b}\right)^{\frac{1}{\delta} k \delta} \approx \frac{\delta}{b} \left(e^{-\frac{1}{b}}\right)^{k \delta} = \frac{\delta}{b} K\left(\frac{k \delta}{b}\right) \quad (23)$$

where  $K(x) := e^{-x}$ . That is  $\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}})$  from (17) is basically the kernel likelihood given in (13) with the one-sided exponential kernel, and  $\hat{\Sigma}_{t_j}$  given by (22) is basically the kernel estimate

$$\hat{\Sigma}_{t_j} = \left[ \sum_k K\left(\frac{k}{bT}\right) \right]^{-1} \sum_k K\left(\frac{k}{bT}\right) \sum_{i=1}^N \omega_{t_j-k}^i (\mathbf{x}_{t_j-k}^i - \mathbf{x}_{t_j-k-1}^i) (\mathbf{x}_{t_j-k}^i - \mathbf{x}_{t_j-k-1}^i)^T.$$

### 3.4 Summary

Our estimation method consists of three components:

- (i) The state-space model with a new market microstructure noise model and the transaction time model for the efficient log-price (Section 3.1);
- (ii) A particle filter which sequentially approximates the filtering distributions of the efficient log-prices given the observed transaction prices (Section 3.2);
- (iii) The on-line EM-type estimator  $\hat{\Sigma}_{t_j}$  given by (21) or (22) which estimates  $\Sigma_{t_j}$  based on the particle approximation of the filtering distribution obtained from the particle filter (Section 3.3). This estimator is improved in the time-varying case to  $\tilde{\Sigma}_{t_j}^*$  in Section 5.

A key aspect of the method is the back and forth between the particle filter and the EM-type estimator. To propagate the particles from time  $t_j$  to time  $t_{j+1}$  the particle filter requires an estimator of  $\Sigma_{t_{j+1}}$  which we denote by  $\hat{\Sigma}_{t_{j+1}}^{\text{pf}}$ . A simple solution is to use  $\hat{\Sigma}_{t_{j+1}}^{\text{pf}} := \hat{\Sigma}_{t_j}$  from the

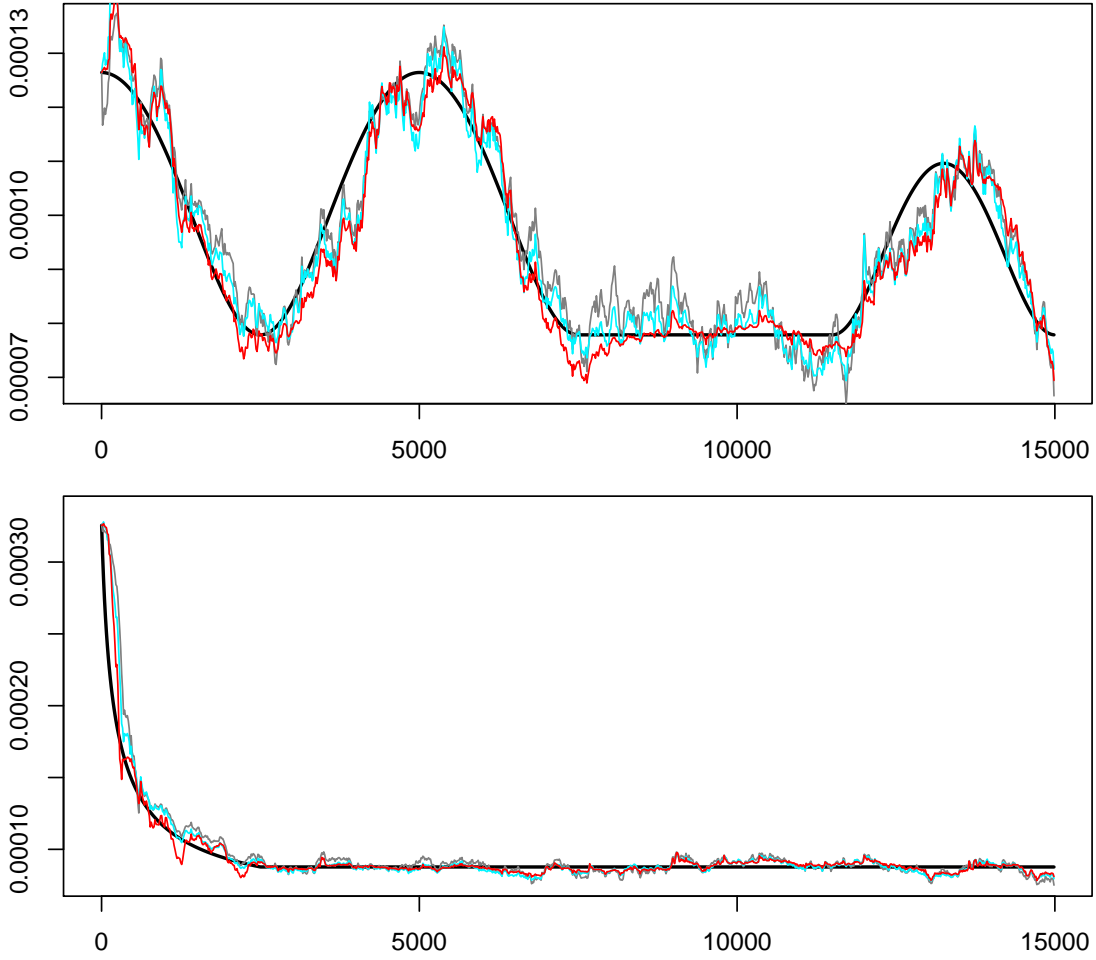


Figure 4: Estimation of two time-varying volatility curves given by the black lines based on simulated data. Estimators:  $\hat{\Sigma}_{t_j}$  (turquoise line),  $\tilde{\Sigma}_{t_j|t_j}^*$  (red line), benchmark estimator (gray line). All estimator use the step sizes (27) where  $\alpha$  and  $\beta$  are optimized for each estimator. For details see Section 7.1.

previous EM-type step. A more sophisticated solution is to use the estimator  $\hat{\Sigma}_{t_{j+1}}^{\text{pf}} := \tilde{\Sigma}_{t_{j+1}|t_j}^*$  from Section 5.2 based on a prediction argument. The EM-type estimator then in turn updates the covariance estimate based on the new particles for time  $t_{j+1}$  generated by the particle filter.

Estimation results of our estimators  $\hat{\Sigma}_{t_j}$ ,  $\tilde{\Sigma}_{t_j|t_j}^*$  (see Section 5), and a benchmark estimator (see Section 7.1) are presented in Figure 4. Details and a discussion are given in Section 7.1.

## 4 From Transaction Time to Clock Time

### 4.1 Clock Time Spot Volatility Estimation

In the preceding section, we have derived an algorithm for the estimation of the covariance matrix  $\Sigma_{t_j} = \Sigma(t_j)$  in a transaction time model. If one prefers a clock time model all results of this paper continue to hold with some modifications. In this case one may consider as the underlying model the stochastic differential equation

$$d\mathbf{X}(t) = \Gamma(t) d\mathbf{W}(t) \quad \text{where} \quad \Gamma(t) \Gamma^T(t) = \Sigma^c(t) \quad (24)$$

and  $\mathbf{W}(t)$  is a multivariate Brownian motion.  $\Sigma^c(t)$  is the volatility curve in the clock time model. Loosely speaking, it denotes volatility per time unit while  $\Sigma(t)$  denotes the volatility per transaction at time  $t$ . The relation between the two curves should be given by (26) (of course this depends on the mathematical definition of  $\Sigma(t)$  and  $\Sigma^c(t)$ ). If we set  $\mathbf{X}_{t_j} = \mathbf{X}(t_j)$  we obtain the same state space model as in (4) and (5) but now with the log-returns  $\mathbf{Z}_{t_j} = \mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}}$  approximately distributed as

$$\mathbf{Z}_{t_j} \sim \mathcal{N}(\mathbf{0}, |t_j - t_{j-1}| \Sigma^c(t_j)).$$

This is the only change needed in the state-space model (4), (5). As an estimate  $\hat{\Sigma}_{t_j}^c$  we can use the on-line estimates (21) and (22) but now with the update matrix  $\check{\Sigma}_{t_j}(\omega_{t_j})$  replaced by

$$\check{\Sigma}_{t_j}^c(\omega_{t_j}^c) := \sum_{i=1}^N \omega_{t_j}^{ci} \frac{(\mathbf{x}_{t_j}^{ci} - \mathbf{x}_{t_{j-1}}^{ci})(\mathbf{x}_{t_j}^{ci} - \mathbf{x}_{t_{j-1}}^{ci})^T}{|t_j - t_{j-1}|} \quad (25)$$

based on the modified filtering particles  $\{\mathbf{x}_{t_{j-1}:j}^{ci}, \omega_{t_j}^{ci}\}_{i=1}^N$ . In Section 5, we discuss bias correction, adaptive and time-varying selection of the step size  $\lambda_j$ , and prediction of future volatilities. All methods can also be applied to  $\Sigma^c(t)$  which is briefly summarized at the end of Section 5.2.

## 4.2 An Alternative Estimator for Clock Time Spot Volatility

In the diffusion model (24) the spot volatility in clock time is

$$\Sigma^c(t) = \lim_{\Delta t \rightarrow 0} \frac{\int_t^{t+\Delta t} \Sigma^c(s) ds}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{\text{Var}(\mathbf{X}(t + \Delta t) - \mathbf{X}(t))}{\Delta t}.$$

To clarify the relation to the transaction time volatility  $\Sigma(t)$  we assume for a moment that the transaction times  $t_j$  are realizations of a stochastic point process with intensity function  $\lambda_I(t)$  (transaction rate) which is independent of the efficient and observed prices. We then have

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \frac{\text{Var}(\mathbf{X}(t + \Delta t) - \mathbf{X}(t))}{\Delta t} &= \lim_{\Delta t \rightarrow 0} \mathbf{E} \frac{\sum_{j: t < t_j \leq t + \Delta t} \Sigma(t_j)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \mathbf{E} \frac{\sum_{j: t < t_j \leq t + \Delta t} \Sigma(t_j)}{|\{j : t < t_j \leq t + \Delta t\}|} \frac{|\{j : t < t_j \leq t + \Delta t\}|}{\Delta t} = \Sigma(t) \lambda_I(t) \end{aligned}$$

that is

$$\Sigma^c(t) = \Sigma(t) \lambda_I(t). \quad (26)$$

We stress that this is primarily a nonparametric relation (“variance per time unit = variance per transaction  $\times$  expected number of transactions per time unit”) and it depends on the underlying model whether this coincides with the definition of  $\Sigma^c(t)$  and  $\Sigma(t)$  given in the model. A model which exactly leads to this formula is the subordinated differential equation  $d\mathbf{X}(t) = \Gamma(t) d\mathbf{W}_{N(t)}$  with a point process  $N(t)$  with intensity  $\lambda_I(t)$  (cf. Howison and Lamper 2001). The unit of  $\Delta t$  (e.g. milliseconds) determines the unit of  $\Sigma(t)$  (e.g. variance per millisecond) and of the intensity (e.g. expected number of transactions per millisecond). An obvious estimate of the clock time volatility therefore is  $\hat{\Sigma}^c(t_j) = \hat{\Sigma}_{t_j} \times |\{\ell : t_j - \Delta t < t_\ell \leq t_j\}| / \Delta t$  with some  $\Delta t$ .



Here we advocate a different estimation method of the intensity function  $\lambda_I(t)$  which is closer related to our on-line scheme, namely the estimation of  $\lambda_I(t)$  by the inverse of the averaged duration times  $\bar{\delta}_j$  defined by the recursion

$$\bar{\delta}_j = (1 - \lambda_j) \bar{\delta}_{j-1} + \lambda_j (t_j - t_{j-1}) \quad \text{with} \quad \bar{\delta}_2 = t_2 - t_1$$

leading with (18) to the alternative clock time volatility estimator

$$\hat{\Sigma}_{\text{alt}}^c(t_j) := \frac{\hat{\Sigma}_{t_j}}{\bar{\delta}_j} = \frac{\sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \check{\Sigma}_{t_{j-k}}(\omega_{t_{j-k}}) + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \check{\Sigma}_{t_2}(\omega_{t_2})}{\sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} (t_{j-k} - t_{j-k-1}) + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] (t_2 - t_1)}$$

(or better with  $\hat{\Sigma}_{t_j}$  replaced by  $\tilde{\Sigma}_{t_j|t_j}^*$  from Section 5). This estimator has a remarkable property: Because  $\check{\Sigma}_{t_\ell}(\omega_{t_\ell}) \approx (t_\ell - t_{\ell-1}) \check{\Sigma}_{t_\ell}^c(\omega_{t_\ell}^c)$  the estimator is of the form

$$\hat{\Sigma}_{\text{alt}}^c(t_j) \approx \frac{\sum_{k=0}^{j-2} w_k \check{\Sigma}_{t_{j-k}}^c(\omega_{t_{j-k}}^c)}{\sum_{k=0}^{j-2} w_k}$$

that is  $\hat{\Sigma}_{\text{alt}}^c(t_j)$  is a weighted average of the  $\check{\Sigma}_{t_\ell}^c(\omega_{t_\ell}^c)$  and therefore also a decent estimator in the clock time model (the “ $\approx$ ” signs stem from the fact that in  $\check{\Sigma}_{t_\ell}(\omega_{t_\ell})$  and  $\check{\Sigma}_{t_\ell}^c(\omega_{t_\ell}^c)$  two different particle filters are used - the effect of this is not clear!). Notice that the denominator  $t_{j-k} - t_{j-k-1}$  in  $\check{\Sigma}_{t_{j-k}}^c(\omega_{t_{j-k}}^c)$  cancels out leading therefore to a more stable estimator (for example the sharp green peaks in Figures 8 and 9 are caused by small values of  $t_{j-k} - t_{j-k-1}$ ).

The above argument contains a pitfall: While  $\Sigma(t)$  usually is smooth thus requiring small values of  $\lambda_j$ , the intensity of the point process  $\lambda_I(t)$  changes considerably over time thus requiring larger values of  $\lambda_j$ . For that reason we use different sequences  $\lambda_j$  for the estimators  $\hat{\Sigma}_{t_j}$  and  $\bar{\delta}_j$ .

The modified estimators  $\tilde{\Sigma}_{t_j|t_j}^{*c}$  (quasi mean squared error corrected version of  $\hat{\Sigma}_{t_j}^c$  as defined in Section 5) and  $\tilde{\Sigma}_{\text{alt}}^c(t_j) := \tilde{\Sigma}_{t_j|t_j}^{*c} / \bar{\delta}_j$  (with the quasi mean squared corrected version  $\tilde{\Sigma}_{t_j|t_j}^{*c}$  instead of  $\hat{\Sigma}_{\text{alt}}^c(t_j)$  - see (42)) are plotted in figures 8 and 9 and discussed in Section 7.2. In this example a constant step size  $\lambda_j \equiv \lambda$  turned out to be sufficient for the estimator  $\bar{\delta}_j$ .

## 5 Fine-Tuning of the Volatility Estimator in the Time-Varying Case

In this section we present a method for the adaptive choice of the time-varying step size  $\lambda_j$  and an on-line bias correction for the estimator  $\hat{\Sigma}_{t_j}$  given by (18) through (19). The basic idea for bias correction is to calculate two estimators with different step sizes in parallel and to balance the two on-line. The resulting estimator is the estimator  $\tilde{\Sigma}_{t_j|t_j}^*$  from Figure 4. We continue to use the notation with  $\Sigma$  although we only discuss the univariate case (the basic formula (35) also holds in the multivariate case with synchronous trading times). We also present an on-line method for quasi mean squared error minimization, and a method for the prediction of future volatilities.

## 5.1 Adaptive Step Size Selection

For constant  $\lambda$  we have the equivalence of the on-line estimator with a kernel estimator with kernel  $K(x) = e^{-x}$  as described in Section 3.3 under 4). For kernel estimators the adaptive (off-line) choice of the bandwidth has been discussed extensively and most of these results could be transferred to the present setting. However, there does not exist any equivalence between our on-line estimator with time-varying  $\lambda_j$  and kernel estimators with local bandwidths: The weight  $\lambda_j$  at time  $t_j$  only applies to the last observation and not to a longer stretch of data.

We are not aware of any rigorous results on adaptive choices for a sequence  $\lambda_j$  for exponential smoothing estimators. This means that the method proposed below may also be of relevance in other on-line estimation settings.

Here is an overview of the method:

1. We start with the ad-hoc proposal based on the logistic function (to ensure  $0 < \lambda_j < 1$ )

$$\lambda_j := \frac{\exp[\alpha + \beta h_{t_{j-1}}]}{1 + \exp[\alpha + \beta h_{t_{j-1}}]}, \quad (27)$$

where

$$h_{t_{j-1}} := \left| \frac{\log \hat{\Sigma}_{t_{j-1}} - \log \hat{\Sigma}_{t_{j-1}}^{(1/2)}}{\frac{j-1}{j-1} - \frac{j-1}{j-1}^{(1/2)}} \right|^2. \quad (28)$$

(27) was proposed by Taylor (2004) with a different  $h_{t_{j-1}}$ . The above  $h_{t_{j-1}}$  is motivated at the end of Section 5.2. For the definition of the expressions in  $h_{t_{j-1}}$  see (33) and below.  $\alpha$  and  $\beta$  are adaptively determined in step 4.

2. At each time  $t_j$  we calculate on-line two different estimators: First  $\hat{\Sigma}_{t_j}$  as defined in (19) and second  $\hat{\Sigma}_{t_j}^{(1/2)}$  which is the same as  $\hat{\Sigma}_{t_j}$  but with all  $\lambda_j$  replaced by  $\lambda_j/2$ . Thus we have at each time step two on-line estimators available - one with a larger step size sequence (with less smoothing) and one with a smaller step size sequence (with stronger smoothing).
3. We then consider arbitrary linear combinations of these estimators and determine at each time step  $t_j$  the optimal linear combination with respect to the optimal quasi mean squared error, or alternatively with respect to unbiasedness resulting in the estimators  $\tilde{\Sigma}_{t_j|t_j}^*$  or  $\check{\Sigma}_{t_j|t_j}$ . The advantage of this method is that it can be performed on-line for each  $t_j$ .
4. The mean squared error of the estimator  $\tilde{\Sigma}_{t_j|t_j}^*$  resulting from the whole procedure 1. through 3. is finally minimized with respect to  $\alpha$  and  $\beta$  by the cross-validation type criterion

$$\text{crit}(\alpha, \beta) := \sum_{j=2}^{T-1} (\tilde{\Sigma}_{t_j|t_j}^* - \check{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}}))^2. \quad (29)$$

This cannot be done on-line. In practice, one will use in an on-line setting the values of  $\alpha$  and  $\beta$  from past experience. The expectation of the above criterion is approximately

$$\sum_{j=2}^{T-1} \left[ (\mathbf{E} \tilde{\Sigma}_{t_j|t_j}^* - \Sigma_{t_j})^2 + \text{Var}(\tilde{\Sigma}_{t_j|t_j}^*) + \text{Var}(\check{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}})) \right].$$

Because the last term does not depend on  $\alpha$  and  $\beta$  we correctly minimize the approximate mean squared error.

We do not know anything about the theoretical properties of the procedure as a whole. We feel however that the degree of adaption is high as a result of the minimization in step 3 (correcting somehow for the limitations of the ad-hoc proposal in step 1) and the final minimization with respect to  $\alpha$  and  $\beta$ . This is confirmed by our simulations.

**Remark:** A simpler alternative is to use a fixed step size  $\lambda_j \equiv \lambda$  and to minimize the mean squared error (29) with respect to  $\lambda$ . Steps 2 and 3 can be kept as they are in this case.

## 5.2 On-line Bias Correction and Mean Squared Error Minimization

We now describe steps 2 and 3 in detail. We stress that these steps can be done for arbitrary step size sequences  $\lambda_j$ , that is we do not need the specific choice from step 1.

Let  $\tau : [0, \infty) \rightarrow [0, \infty)$  be the mapping that maps transaction time to clock time, i.e.  $\tau(j) = t_j$  (we assume that  $\tau$  is defined on the whole positive real line). We define

$$\dot{\Sigma}(s) := \frac{\partial}{\partial s} \Sigma(\tau(s)) = \Sigma'(\tau(s)) \tau'(s)$$

leading to the linear approximation

$$\Sigma(t_j) = \Sigma(\tau(j)) \approx \Sigma(t_i) + (j - i) \dot{\Sigma}(i) \quad (30)$$

(for the meaning of the “ $\approx$ ”-sign see Section 8; for example in the equidistant case  $t_i = i\delta$  we have  $\tau'(i) = \delta$  and  $(j - i) \dot{\Sigma}(i) = (j - i) \delta \Sigma'(t_i)$  is small for small  $\delta$ ). By using the approximation

$$\begin{aligned} \mathbf{E} \check{\Sigma}_{t_j}(\omega_{t_j}) &= \mathbf{E} \sum_{i=1}^N \omega_{t_j}^i (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T \approx \mathbf{E}[\mathbf{E}((\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^T | \mathbf{Y}_{t_{1:j}})] \\ &= \mathbf{E}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^T = \Sigma_{t_j} \end{aligned} \quad (31)$$

(compare the discussion at the end of 2) in Section 3.3) we obtain from (18) (with some  $i$  close to  $j$ )

$$\begin{aligned} \mathbf{E} \hat{\Sigma}_{t_j} &\approx \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \Sigma(t_{j-k}) + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \Sigma(t_2) \\ &\approx \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \left[ \Sigma(t_i) - (i - (j - k)) \dot{\Sigma}(i) \right] + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \left[ \Sigma(t_i) - (i - 2) \dot{\Sigma}(i) \right] \\ &= \Sigma(t_i) - (i - \bar{j}) \dot{\Sigma}(i) \approx \Sigma(t_i) - (i - \bar{j}) \dot{\Sigma}(\bar{j}) \approx \Sigma(t_{\bar{j}}) \end{aligned} \quad (32)$$

with

$$\bar{j} := \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} (j - k) + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] 2. \quad (33)$$

We note that  $\bar{j}$  can be obtained via the on-line recursion

$$\bar{j} = (1 - \lambda_j) \overline{j-1} + \lambda_j j \quad \text{with} \quad \bar{2} = 2. \quad (34)$$

This means that we are estimating  $\Sigma(t)$  essentially at time  $t_{\bar{j}} < t_j$ . This is a result of the one-sidedness of the recursive method (for example in the equidistant case  $t_j = j\delta$  and  $\lambda_j \equiv \lambda$  we obtain  $\bar{j} \approx j + 1 - 1/\lambda$  and  $t_{\bar{j}} \approx (j + 1 - 1/\lambda)\delta$  - see also (23)). In order to correct for this bias or to construct even approximately unbiased estimators of future volatilities we now take a linear combination of the two estimators  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  where the latter is the same as  $\hat{\Sigma}_{t_j}$  in (18) and (19) but with all  $\lambda_j$  replaced by  $\lambda_j/2$ . Analogously we define  $\bar{j}^{(1/2)}$  as in (33) and (34) but again with all  $\lambda_j$  replaced by  $\lambda_j/2$ . The new estimator now is defined by the extrapolation

$$\tilde{\Sigma}_{t_i|t_j} := (1 + \kappa_{i|j}) \hat{\Sigma}_{t_j} - \kappa_{i|j} \hat{\Sigma}_{t_j}^{(1/2)} \quad (35)$$

with time-varying weights

$$\kappa_{i|j} := \frac{i - \bar{j}}{\bar{j} - \bar{j}^{(1/2)}}. \quad (36)$$

We immediately obtain

$$\mathbf{E} \tilde{\Sigma}_{t_i|t_j} \approx \Sigma(t_i) - \left[ (1 + \kappa_{i|j}) (i - \bar{j}) - \kappa_{i|j} (i - \bar{j}^{(1/2)}) \right] \dot{\Sigma}(i) = \Sigma(t_i)$$

and for  $i = j$  we therefore have a bias-corrected estimator of  $\Sigma(t_j)$ . Because the estimator extrapolates the two estimators  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  we have to watch particularly the variance which may become large. From a statistical view a better choice is the estimator with a minimal mean squared error. In the appendix, we calculate the quasi mean squared error (with the unknown efficient log-prices used instead of the filter particles) and show that this mean squared error is minimized by

$$\kappa_{\min} \approx \frac{(i - \bar{j})(\bar{j} - \bar{j}^{(1/2)}) \left[ \frac{\partial}{\partial t} \log \Sigma(t)|_{t=t_{\bar{j}}} \tau'(\bar{j}) \right]^2 - 2(v_{1,j} - v_{3,j})}{(\bar{j} - \bar{j}^{(1/2)})^2 \left[ \frac{\partial}{\partial t} \log \Sigma(t)|_{t=t_{\bar{j}}} \tau'(\bar{j}) \right]^2 + 2(v_{1,j} + v_{2,j} - 2v_{3,j})}$$

with  $v_{1,j}$ ,  $v_{2,j}$  and  $v_{3,j}$  obtained from the recursions

$$v_{1,j} = (1 - \lambda_j)^2 v_{1,j-1} + \lambda_j^2, \quad v_{1,2} = 1; \quad (37)$$

$$v_{2,j} = \left(1 - \frac{\lambda_j}{2}\right)^2 v_{2,j-1} + \frac{\lambda_j^2}{4}, \quad v_{2,2} = 1; \quad (38)$$

$$v_{3,j} = (1 - \lambda_j) \left(1 - \frac{\lambda_j}{2}\right) v_{3,j-1} + \frac{\lambda_j^2}{2}, \quad v_{3,2} = 1. \quad (39)$$

$\frac{\partial}{\partial t} \log \Sigma(t)|_{t=t_{\bar{j}}}$  and  $\tau'(\bar{j})$  are unknown. In order to get an adaptive choice of  $\kappa$  we replace these terms by estimators. From (32) we know that  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  are essentially estimators of  $\Sigma(t)$  at times  $t_{\bar{j}}$  and  $t_{\bar{j}^{(1/2)}}$ , respectively. We therefore use

$$\frac{\log \hat{\Sigma}_{t_j} - \log \hat{\Sigma}_{t_j}^{(1/2)}}{t_{\bar{j}} - t_{\bar{j}^{(1/2)}}} \frac{t_{\bar{j}} - t_{\bar{j}^{(1/2)}}}{\bar{j} - \bar{j}^{(1/2)}} \quad (40)$$

as an estimate of  $\frac{\partial}{\partial t} \log \Sigma(t)|_{t=t_{\bar{j}}} \tau'(\bar{j})$  leading to

$$\kappa_{i|j}^* := \frac{\frac{i - \bar{j}}{\bar{j} - \bar{j}^{(1/2)}} \left[ \log \hat{\Sigma}_{t_j} - \log \hat{\Sigma}_{t_j}^{(1/2)} \right]^2 - 2(v_{1,j} - v_{3,j})}{\left[ \log \hat{\Sigma}_{t_j} - \log \hat{\Sigma}_{t_j}^{(1/2)} \right]^2 + 2(v_{1,j} + v_{2,j} - 2v_{3,j})} \quad (41)$$

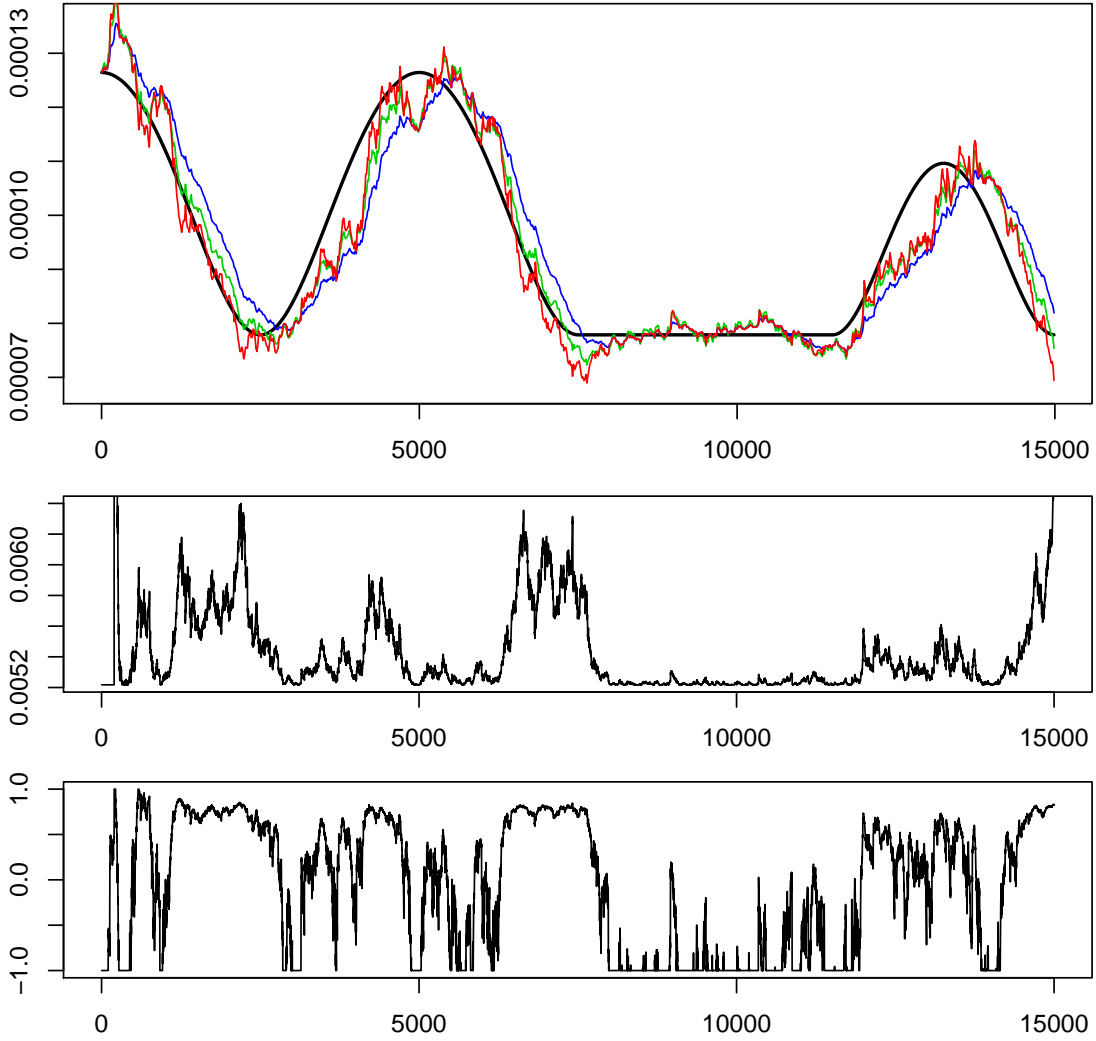


Figure 5: Estimation of the time-varying volatility curve given by the black line in the upper plot based on simulated data. Upper plot:  $\hat{\Sigma}_{t_j|t_j}^*$  (red line),  $\hat{\Sigma}_{t_j}$  (green line),  $\hat{\Sigma}_{t_j}^{(1/2)}$  (blue line) (all with the same step sizes given by (27)); middle plot: step size sequence  $\lambda_j$ ; lower plot: sequence  $\kappa_{i|j}^*$ . For details see Section 7.1.

and the corresponding estimator

$$\tilde{\Sigma}_{t_i|t_j}^* := (1 + \kappa_{i|j}^*) \hat{\Sigma}_{t_j} - \kappa_{i|j}^* \hat{\Sigma}_{t_j}^{(1/2)}. \quad (42)$$

In practice, the values of  $\kappa_{i|j}^*$  will be restricted to the interval  $[-1, 1]$  because other values do not make sense (smaller values than  $-1$  may occur because  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  are correlated - however such values yield an extrapolation in the wrong time direction).

An example of this estimator for simulated data is given in Figure 5. The bias of  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  and the bias correction of  $\tilde{\Sigma}_{t_j|t_j}^*$  are clearly visible. For details see Section 7.1. The estimators look slightly undersmoothed. We comment on that at the end of Section 7.1.

It is easy to prove that  $(v_{1,j} + v_{2,j} - 2v_{3,j}) \geq 0$ . “Usually” also  $v_{1,j} - v_{3,j} \geq 0$  (for example for constant  $\lambda_j \equiv \lambda$   $v_{1,j}$  and  $v_{3,j}$  converge to the fixpoints of the recursion  $v_1 = \frac{\lambda}{2-\lambda}$  and  $v_3 = \frac{\lambda}{3-\lambda}$  with  $v_{1,j} - v_{3,j} > 0$ ). For this reason we usually have  $\kappa_{i|j}^* < \kappa_{i|j}$ .

For the recursion described in Section 3.4 (where  $\Sigma(t_{j+1})$  is needed in the next step of the particle filter) we think that the mean squared error choice  $\tilde{\Sigma}_{t_{j+1}|t_j}^*$  with  $\kappa_{j+1|j}^*$  is the best choice.

On the contrary as an estimate for  $\Sigma(t_j)$  of financial log-returns the unbiased estimator with  $\kappa_{j|j}$  may be more interesting (it is less smoothed and contains in some sense more information). Perhaps in a practical application both estimators (with  $\kappa_{j|j}$  and  $\kappa_{j|j}^*$ ) should be plotted.

We finally motivate the choice of  $\lambda_j$  and  $h_{t_{j-1}}$  in step 1: In the case of constant  $\lambda_j = \lambda$  we obtain from (32) and (47) for the mean squared error

$$\mathbf{E}\left(\hat{\Sigma}_{t_j} - \Sigma(t_j)\right)^2 \approx 1/\lambda^2 \dot{\Sigma}(\bar{j})^2 + \lambda \Sigma(t_j)^2$$

which gets minimal for

$$\lambda = \left| \sqrt{2} \frac{\partial}{\partial t} \log \Sigma(t)|_{t=\bar{j}} \tau'(\bar{j}) \right|^{2/3}.$$

Together with the restriction  $0 < \lambda_j < 1$  (leading to the use of the logistic function) and the estimate (40) this has motivated the local choice of  $\lambda_j$  as in (27) with

$$h_{t_{j-1}} := \left| \frac{\log \hat{\Sigma}_{t_{j-1}} - \log \hat{\Sigma}_{t_{j-1}}^{(1/2)}}{\bar{j} - 1 - \bar{j} - 1^{(1/2)}} \right|^\rho$$

where  $\alpha$  and  $\beta$  are determined as described in step 4. We have simulated the mean squared error of the whole procedure 1. through 4. for several values of  $\rho$  leading finally to the choice  $\rho = 2$  as in (28). Nevertheless, the choice of  $\lambda_j$  and  $h_{t_{j-1}}$  as given in (27) and (28) remains to be an ad-hoc suggestion.

**Bias correction in clock time models:** A similar algorithm for adaption and bias correction can be established in the clock time setting from Section 4. Instead of the approximation (30) we start with

$$\Sigma^c(t_j) = \Sigma^c(t_i) + (t_j - t_i) \Sigma^{c'}(t_i)$$

and define instead of  $\bar{j}$

$$\bar{t}_j := \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} t_{j-k} + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] t_2$$

given by the on-line recursion

$$\bar{t}_j = (1 - \lambda_j) \bar{t}_{j-1} + \lambda_j t_j \quad \text{with} \quad \bar{t}_2 = t_2.$$

Analogously we obtain the estimator

$$\tilde{\Sigma}_{t_i|t_j}^c := (1 + \kappa_{t_i|t_j}) \hat{\Sigma}_{t_j}^c - \kappa_{t_i|t_j} \hat{\Sigma}_{t_j}^{c(1/2)}$$

with

$$\kappa_{t_i|t_j} := \frac{t_i - \bar{t}_j}{\bar{t}_j - \bar{t}_j^{(1/2)}}$$

as the approximately unbiased estimator and  $\tilde{\Sigma}_{t_i|t_j}^{*c}$  with

$$\kappa_{t_i|t_j}^* \approx \frac{\frac{t_i - \bar{t}_j}{\bar{t}_j - \bar{t}_j^{(1/2)}} \left[ \log \hat{\Sigma}_{t_j}^c - \log \hat{\Sigma}_{t_j}^{c(1/2)} \right]^2 - 2(v_{1,j} - v_{3,j})}{\left[ \log \hat{\Sigma}_{t_j}^c - \log \hat{\Sigma}_{t_j}^{c(1/2)} \right]^2 + 2(v_{1,j} + v_{2,j} - 2v_{3,j})}$$

as the estimator with approximately optimal quasi mean squared error.

**Prediction:** The estimators  $\tilde{\Sigma}_{t_i|t_j}$  and  $\tilde{\Sigma}_{t_i|t_j}^*$  can be used (with  $i > j$ ) for prediction of future volatilities. In particular in combination with a predictor for future durations (e.g. with an ACD model - cf. Engle and Russell (1998)) this may lead to new predictors. One should keep in mind that these predictions are based on linear extrapolation. However, it should be possible to adapt the methods of this paper also to other prediction models such as in Meddahi et al. (2006). By plugging the relation

$$\frac{t_i - \bar{t}_j}{\bar{t}_j - \bar{t}_j^{(1/2)}} \approx \frac{(i - \bar{j}) \tau'(\bar{j})}{(\bar{j} - \bar{j}^{(1/2)}) \tau'(\bar{j})} = \frac{i - \bar{j}}{\bar{j} - \bar{j}^{(1/2)}}$$

into (36) and (41) and replacing afterwards  $t_i$  by  $t$  we can also obtain predictors for arbitrary time points  $t$ . Similarly the above estimators from the clock time model can be used for prediction.

## 6 Implementation Overview in the Time-Varying Case

### The Algorithm

For transaction time we use the algorithm as described in Section 3.4 with  $\hat{\Sigma}_{t_j}^{\text{pf}} = \tilde{\Sigma}_{t_j|t_{j-1}}^*$  in the update step of the particle filter. As the estimator of  $\Sigma_{t_j}$  we usually use  $\tilde{\Sigma}_{t_j|t_j}^*$  from (42) (if not otherwise stated) and sometimes  $\tilde{\Sigma}_{t_j|t_j}$  from (35). This means in particular that we are applying the adaptation procedure described in Section 5.1.  $\alpha$  and  $\beta$  are used from past experience or determined as described in step 4.

The particle filter uses the following steps for  $j = 2, \dots, T$  (see Proposition 1)

- For  $i = 1, \dots, N$ :
  - Generate  $\mathbf{x}_{t_j}^i$  from the optimal proposal  $\mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i; \hat{\Sigma}_{t_j}^{\text{pf}}) |_{\log \mathbf{A}_{t_j}}$ .
  - Compute the importance weight  $\check{\omega}_{t_j}^i$  as in (7). If  $S = 1$  this is given by

$$\check{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \left\{ \Phi(\sup \log A_{t_j} | x_{t_{j-1}}^i; \hat{\Sigma}_{t_j}^{\text{pf}}) - \Phi(\inf \log A_{t_j} | x_{t_{j-1}}^i; \hat{\Sigma}_{t_j}^{\text{pf}}) \right\}.$$

- For  $i = 1, \dots, N$ : Normalize the importance weight  $\omega_{t_j}^i = \check{\omega}_{t_j}^i / \sum_{k=1}^N \check{\omega}_{t_j}^k$ .
- If the effective sample size  $\text{ESS}(\{\omega_{t_j}^i\}_{i=1}^N) < 0.2N$ , then resample the particles using, for instance, the residual resampling scheme (Douc et al. 2005).

The whole algorithm is computationally very efficient because the complexity of one iteration is linear in the number of particles  $N$ . As a result of the efficiency of our particle filter, the number of particles  $N$  is not a critical quantity. Typically, about 500 particles suffice to achieve a reasonable precision (see Figure 6).

### Initialization

Our experience from many data sets is that the algorithm stabilizes quickly provided that reasonable starting values are used – e.g.  $\hat{\Sigma}_{t_2} = \hat{\Sigma}_{t_2}^{(1/2)} = \hat{\Sigma}_{t_2}^{\text{pf}} = \hat{\Sigma}$  with  $\hat{\Sigma}$  from prior knowledge or with

$\hat{\Sigma}$  being a rough initial estimate. The particle filter is started by simulating the  $x_{t_1,s}^i$  such that the  $\exp[x_{t_1,s}^i]$  are uniformly distributed on  $A_{t_1,s}$ . In that case one will use  $v_{1,3} = \frac{\lambda_3}{2-\lambda_3}$ ,  $v_{2,3} = \frac{\lambda_3}{4-\lambda_3}$  and  $v_{3,3} = \frac{\lambda_3}{3-\lambda_3}$  with (say)  $\lambda_3 = 1/500$  (these are the fix points of the recursions (37) through (39)).

More sophisticated starting values are obtained as follows: One uses the procedure of this paper over the first 500 transactions in reversed time order leading to values  $\hat{\Sigma}_{t_1}^{\text{rev}}$ ,  $\hat{\Sigma}_{t_1}^{\text{rev}(1/2)}$ ,  $\bar{\Gamma}^{\text{rev}}$ ,  $\bar{\Gamma}^{\text{rev}(1/2)}$  and starts the algorithm then with the following values obtained by extrapolation:

$$\begin{aligned}\bar{2} &:= -\bar{\Gamma}^{\text{rev}} + 3; & \bar{2}^{(1/2)} &:= -\bar{\Gamma}^{\text{rev}(1/2)} + 3; \\ \hat{\Sigma}_{t_2} &:= \left(1 + \frac{2 \times \bar{\Gamma}^{\text{rev}} - 2}{\bar{\Gamma}^{\text{rev}(1/2)} - \bar{\Gamma}^{\text{rev}}}\right) \hat{\Sigma}_{t_1}^{\text{rev}} - \frac{2 \times \bar{\Gamma}^{\text{rev}} - 2}{\bar{\Gamma}^{\text{rev}(1/2)} - \bar{\Gamma}^{\text{rev}}} \hat{\Sigma}_{t_1}^{\text{rev}(1/2)}; \\ \hat{\Sigma}_{t_2}^{(1/2)} &:= \left(1 + \frac{\bar{\Gamma}^{\text{rev}} + \bar{\Gamma}^{\text{rev}(1/2)} - 2}{\bar{\Gamma}^{\text{rev}(1/2)} - \bar{\Gamma}^{\text{rev}}}\right) \hat{\Sigma}_{t_1}^{\text{rev}} - \frac{\bar{\Gamma}^{\text{rev}} + \bar{\Gamma}^{\text{rev}(1/2)} - 2}{\bar{\Gamma}^{\text{rev}(1/2)} - \bar{\Gamma}^{\text{rev}}} \hat{\Sigma}_{t_1}^{\text{rev}(1/2)};\end{aligned}$$

$v_{1,2} = \frac{\lambda_1^{\text{rev}}}{2-\lambda_1^{\text{rev}}}$ ;  $v_{2,2} = \frac{\lambda_1^{\text{rev}}}{4-\lambda_1^{\text{rev}}}$ ;  $v_{3,2} = \frac{\lambda_1^{\text{rev}}}{3-\lambda_1^{\text{rev}}}$ . We then obtain e.g. for the bias-corrected estimator (where  $\kappa_{2|2} = \frac{2-\bar{2}}{2-\bar{2}^{(1/2)}} = \frac{\bar{\Gamma}^{\text{rev}}-1}{\bar{\Gamma}^{\text{rev}(1/2)}-\bar{\Gamma}^{\text{rev}}} = \kappa_{1|1}^{\text{rev}}$ ) after some calculations

$$\begin{aligned}\tilde{\Sigma}_{t_2|t_2} &= (1 + \kappa_{2|2}) \hat{\Sigma}_{t_2} - \kappa_{2|2} \hat{\Sigma}_{t_2}^{(1/2)} = \dots = \\ &= \left(1 + \frac{\bar{\Gamma}^{\text{rev}} - 1}{\bar{\Gamma}^{\text{rev}(1/2)} - \bar{\Gamma}^{\text{rev}}}\right) \hat{\Sigma}_{t_1}^{\text{rev}} - \frac{\bar{\Gamma}^{\text{rev}} - 1}{\bar{\Gamma}^{\text{rev}(1/2)} - \bar{\Gamma}^{\text{rev}}} \hat{\Sigma}_{t_1}^{\text{rev}(1/2)} = \hat{\Sigma}_{t_1|t_1}^{\text{rev}}\end{aligned}$$

(note that because of  $\mathbf{X}_{t_1} - \mathbf{X}_{t_2} = -\mathbf{Z}_{t_2}$  with  $\mathbf{Z}_{t_2} \sim \mathcal{N}(\mathbf{0}, \Sigma_{t_2})$  we have  $\Sigma_{t_1}^{\text{rev}} = \Sigma_{t_2}$ ). The particle filter could be started with  $\hat{\Sigma}_{t_2}^{\text{pf}} := \tilde{\Sigma}_{t_2|t_2}$ .  $\lambda_3$  can then be calculated from the above formulas.

The reversed method works nicely as can be seen from Figure 7 below.

In order to exclude the effect of starting values we have used in the simulations (except from Figure 6) the true matrix  $\Sigma_{t_2}$  as the starting value (i.e.  $\hat{\Sigma}_{t_2}^{\text{pf}} = \hat{\Sigma}_{t_2} = \hat{\Sigma}_{t_2}^{(1/2)} = \Sigma_{t_2}$ ).

## 7 Simulations and Applications

### 7.1 Results for Simulated Data

#### Estimation of time-constant spot volatility

We first consider the estimation of time-constant spot volatility. An efficient log-price process is simulated from  $t_1$  to  $t_{5000}$  with squared volatility equal to  $\Sigma_t = 0.0001^2$ . The initial efficient price  $\exp[X_{t_1}]$  is sampled from a uniform distribution on  $[50 - 0.005, 50 + 0.005]$ . The transaction prices are obtained by rounding the efficient prices to the nearest cent which constitutes a special case of our market microstructure noise model. Our algorithm for time-constant spot volatility estimation (21) is applied with different numbers of particles  $N$  and different values of  $\gamma$ . The starting value  $\hat{\Sigma}_{t_2}^{\text{pf}} = \hat{\Sigma}_{t_2}$  is drawn from a uniform distribution on  $(0.00009^2, 0.00011^2)$ . For comparison the results of two benchmark algorithms are also reported. The first benchmark method (“Benchmark” in Figure 6) is a recursive estimator with a simpler microstructure noise correction. It is related to the method in Zumbach et al. (2002) and it is based on the market microstructure model  $\log Y_{t_j} = X_{t_j} + U_{t_j}$ , where the noise variables  $U_{t_j}$  are i.i.d. with  $\text{Var } U_{t_j} = \eta^2$ . The recursive



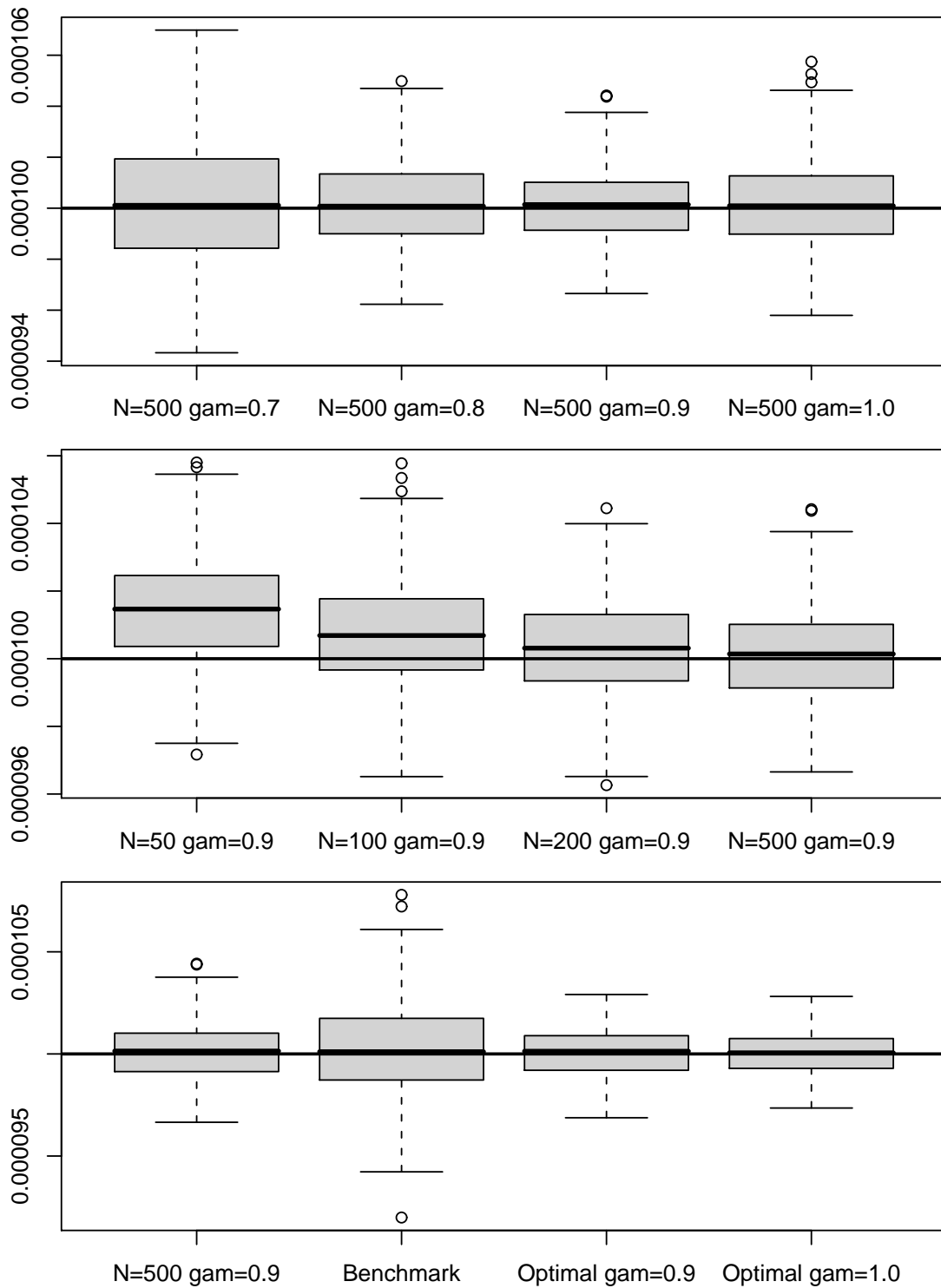


Figure 6: Box plots for the estimation of a time-constant volatility based on simulated data (5,000 transactions). The estimator (21) is applied with different numbers of particles  $N$  and different  $\gamma$  and compared to the benchmark estimator and the optimal estimator (not available in practice). The box plots are based on 500 independent runs.

estimator is given by

$$\hat{\Sigma}_{t_j}^B := \left\{1 - \frac{1}{j-1}\right\} (\hat{\Sigma}_{t_{j-1}}^B + \max\{0, 2\hat{\eta}_{t_{j-1}}^2\}) + \frac{1}{j-1} (\log y_{t_j} - \log y_{t_{j-1}})^2 - \max\{0, 2\hat{\eta}_{t_j}^2\} \quad (43)$$

where  $\hat{\eta}_{t_j}^2 := \left\{1 - \frac{1}{j-2}\right\} \hat{\eta}_{t_{j-1}}^2 - \frac{1}{j-2} (\log y_{t_j} - \log y_{t_{j-1}}) (\log y_{t_{j-1}} - \log y_{t_{j-2}})$  (here  $\frac{1}{j-2}$  is used instead of  $\frac{1}{j-1}$  because the algorithm starts one time point later). The term  $\max\{0, 2\hat{\eta}_{t_j}^2\}$  corrects for the market microstructure noise. This follows from the fact that

$$\text{Cov}(\log Y_{t_j} - \log Y_{t_{j-1}}, \log Y_{t_{j-1}} - \log Y_{t_{j-2}}) = -\eta^2.$$

The second benchmark method is, in some sense, the optimal estimator (“Optimal” in Figure 6). It is unavailable in practice because it uses the latent efficient log-prices. It is computed analogous to (21) but instead of the particles it employs the efficient log-prices leading to

$$\hat{\Sigma}_{t_j}^{\text{Opt}} = \{1 - (j-1)^{-\gamma}\} \hat{\Sigma}_{t_{j-1}}^{\text{Opt}} + (j-1)^{-\gamma} (x_{t_j} - x_{t_{j-1}})^2.$$

The simulation results are given in terms of box plots which are obtained by 500 independent runs (Figure 6). The box plots suggest that our volatility estimator is asymptotically unbiased and that  $\gamma = 0.9$  is a reasonable value. We can also conclude that about 500 particles are sufficient which makes our algorithm computationally efficient and suitable for real-time applications. In addition, it can be observed that the benchmark estimator has a larger variance than our estimator.

### Estimation of time-varying spot volatility

We now compare our algorithms (22) and (42) for the time-varying spot volatility estimators  $\hat{\Sigma}_{t_j}$  and  $\tilde{\Sigma}_{t_j|t_j}^*$ , respectively, with a benchmark estimator. The efficient log-prices are generated with respect to the time-varying volatility given by the black lines in Figure 4. The first case (upper plot) is more challenging while the second case (lower plot) is more realistic for a volatility curve in transaction time - see the real data example in Figure 7. In both cases  $\exp[X_{t_1}] \sim \mathcal{U}[50 - 0.005, 50 + 0.005]$ . Again transaction prices (observations) are obtained by rounding the efficient prices to the nearest cent. 15,000 transactions are generated which is typical for one trading day of a liquid stock. The particle filter is applied with  $N = 500$  particles. The estimator  $\tilde{\Sigma}_{t_j|t_j}^*$  is calculated as described in Sections 5 and 6.  $\hat{\Sigma}_{t_j}$  also uses the time-varying step sizes (27) where  $\alpha$  and  $\beta$  are obtained by minimizing the criterion (29) as for  $\tilde{\Sigma}_{t_j|t_j}^*$ . (A simpler strategy avoiding the calculation of  $\hat{\Sigma}_{t_j}^{(1/2)}$  is to use a constant step size  $\lambda$  obtained by minimizing (29).) Analogous to (43) we consider the benchmark estimator given by

$$\hat{\Sigma}_{t_j}^B := \{1 - \lambda_j\} (\hat{\Sigma}_{t_{j-1}}^B + \max\{0, 2\hat{\eta}_{t_{j-1}}^2\}) + \lambda_j (\log y_{t_j} - \log y_{t_{j-1}})^2 - \max\{0, 2\hat{\eta}_{t_j}^2\} \quad (44)$$

with  $\hat{\eta}_{t_j}^2 := \left\{1 - \frac{1}{j-2}\right\} \hat{\eta}_{t_{j-1}}^2 - \frac{1}{j-2} (\log y_{t_j} - \log y_{t_{j-1}}) (\log y_{t_{j-1}} - \log y_{t_{j-2}})$ . For a fair comparison we also use the time-varying step sizes (27) where  $\alpha$  and  $\beta$  are obtained by minimizing the criterion

$$\sum_{j=2}^{T-1} (\hat{\Sigma}_{t_j}^B + \max\{0, 2\hat{\eta}_{t_j}^2\} - (\log y_{t_{j+2}} - \log y_{t_{j+1}})^2)^2 \quad (45)$$

(the terms  $\hat{\Sigma}_{t_j}^B + \max\{0, 2\hat{\eta}_{t_j}^2\}$  and  $(\log y_{t_{j+2}} - \log y_{t_{j+1}})^2$  are independent in the additive microstructure noise model  $\log Y_{t_j} = X_{t_j} + U_{t_j}$  with  $U_{t_j}$  i.i.d. - thus by using  $(\log y_{t_{j+2}} - \log y_{t_{j+1}})^2$  (45) becomes a decent estimate of the mean squared error (plus a term constant in  $\alpha$  and  $\beta$ )). For  $\hat{\eta}_{t_j}^2$  we use the step sizes  $\frac{1}{j-2}$  because  $\eta_t^2$  should be close to a constant function.

All estimators use the true volatility as starting value. Typical outcomes of the estimators are given in Figure 4. Note that volatility is plotted (instead of squared volatility). Because the true  $\Sigma(t_j)$  is known we can compute the mean squared error  $\sum_{j=2}^{T-1} (\hat{\Sigma}(t_j) - \Sigma(t_j))^2$  for all estimators which gives  $1.21 \times 10^{-18}$ ,  $1.14 \times 10^{-18}$ , and  $1.34 \times 10^{-18}$  (upper plot in Figure 4) and  $3.20 \times 10^{-19}$ ,  $1.77 \times 10^{-19}$ , and  $6.76 \times 10^{-19}$  (lower plot in Figure 4) for the estimators  $\hat{\Sigma}_{t_j}$ ,  $\tilde{\Sigma}_{t_j|t_j}^*$ , and  $\hat{\Sigma}_{t_j}^B$ , respectively. In both plots,  $\tilde{\Sigma}_{t_j|t_j}^*$  significantly outperforms the other estimators.

We have tried to improve the benchmark estimator by a bias correction similar to Section 5. Surprisingly, this has lead only to minor improvements. (We have refrained from plotting this estimator.) The reason for this is not clear: We think that the rounding in the values  $y_{t_j}$  is responsible for the bad quality in that it leads to a (local) bias and higher fluctuations. Perhaps the estimator may be improved a bit by modifying (28).

To further investigate the estimator  $\tilde{\Sigma}_{t_j|t_j}^*$  we have also plotted in Figure 5  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  from (42) (i.e. all estimators with the same  $\alpha$  and  $\beta$  used to optimize  $\tilde{\Sigma}_{t_j|t_j}^*$ ) as well as the sequences  $\lambda_j$  and  $\kappa_{j|j}^*$ . The bias of  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^{(1/2)}$  is clearly visible. Furthermore, it can be seen how the estimator  $\tilde{\Sigma}_{t_j|t_j}^*$  extrapolates these raw estimates to improve on the bias. During the period of constant volatility the step size  $\lambda_j$  gets low because (28) is close to zero. Furthermore,  $\kappa_{j|j}^*$  gets close to  $-1$  which implies  $\tilde{\Sigma}_{t_j|t_j}^* \approx \hat{\Sigma}_{t_j}^{(1/2)}$  (which is the smoother estimate). During periods of volatility changes the step size  $\lambda_j$  gets large and  $\tilde{\Sigma}_{t_j|t_j}^*$  adapts more quickly to  $\Sigma_{t_j}$ .

The general impression from figures 4 and 5 is that the estimators are undersmoothed. This is in part due to the on-line procedure which (for constant  $\lambda$ ) corresponds to a one-sided kernel. Additional variability comes in from the particle filter where the estimated covariance matrix is used instead of the true one. The third point is a limitation of the adaptation procedure from Section 5 where the smallest possible  $\lambda_j$  is  $\frac{1}{1+\exp(-\alpha)}$ . It is not very difficult to improve on that (e.g. by using  $\hat{\Sigma}_{t_j}^{(1/4)}$  with step-sizes  $\lambda_j/4$  instead of  $\hat{\Sigma}_{t_j}^{(1/2)}$ ). The price to pay are estimates which react much slower to changes of the volatility. Thus we feel that our estimates come close to the need of practitioners who like to have a quickly reacting estimate and who prefer to correct undersmoothed estimates by "eye inspection".

## 7.2 Results for Real Data

We use stock data from the TAQ data base. Transactions and market maker quotes of the symbol C (Citigroup) for the 3rd September 2007 were extracted from the TAQ data base. To improve the data quality we carried out the following data cleaning and transformation.

**Cleaning A:** Delete all transactions (quotes) with time stamps outside the main trading period (9:30 AM to 4 PM).

**Cleaning B:** Delete all transactions (quotes) that are not originating from the NYSE.

**Cleaning C:** Delete all transactions with abnormal sale condition or corrected prices (see the TAQ User's Guide for details).

**Data transformation:** If multiple transactions have the same time stamp (after the data cleaning) apply the following transformation. Assume  $t_j = t_{j+1} = \dots = t_{k-1} \neq t_k$ . Replace  $t_l$  by  $t'_l = t_j + (l - j)(t_k - t_j)/(k - j)$  for  $l = j + 1, \dots, k - 1$ .

After the data cleaning 16,287 transactions remained. The transformation replaces identical time stamps with time stamps that are equally spaced. This transformation is necessary because the time stamp precision of our data is limited to one second. We mention that the transformation is only required for clock time volatility estimation. If the volatility is estimated in transaction time then the times stamps are irrelevant (only the order of the transactions matters).

Unfortunately, the quality of the TAQ data is too poor to match easily the transactions with the market maker quotes. Note that it is necessary for our method that the transaction and quote data are perfectly matched. Therefore, our simulations are mainly focused on transaction data.

### Estimation results for real market maker quotes

In order to show how our method works in the case when market maker quotes are available (case 2 in Section 2) we matched by hand (through an adjustment of the time stamps) the quotes and transactions of symbol C for a fraction of the trading day. As mentioned earlier, the quality of our data is too poor to do this automatically. Our particle filter is used with  $N = 5,000$  particles to estimate the filtering distributions of the unknown efficient (log-)prices. Figure 1 gives kernel density estimates of filtering distributions of some efficient prices which are computed based on the particle approximations. The market maker quotes, the transaction prices, and supports of the filtering distributions are also shown. From the figure it can be seen that some filtering distributions are highly skewed. In addition, consecutive zero returns lead to very uninformative filtering distributions (see transactions 2,300 through 2,309).

### Estimation results for real transaction data

We apply our estimators  $\hat{\Sigma}_{t_j}$  and  $\tilde{\Sigma}_{t_j|t_j}^*$  with  $N = 500$  particles and the benchmark method  $\hat{\Sigma}_{t_j}^B$  (44) to estimate the spot volatility for C. To obtain a good initialization for the estimator  $\tilde{\Sigma}_{t_j|t_j}^*$  the initialization algorithm which proceeds in reversed time order is applied to the first 500 transactions (see Section 6). For  $\hat{\Sigma}_{t_j}$  and  $\hat{\Sigma}_{t_j}^B$  an initial volatility of 0.0005 is used.

The transaction data of C and the volatility estimators are shown in Figure 7. At the beginning of the trading day the volatility is large and highly varying. Later, the volatility settles down and seems to be almost constant. Therefore, the localized step size selection from (27) is clearly advantageous compared to fixed step sizes. Again the benchmark estimator is rougher than our estimators. Practically, the volatility in transaction time is almost constant after 10:00.

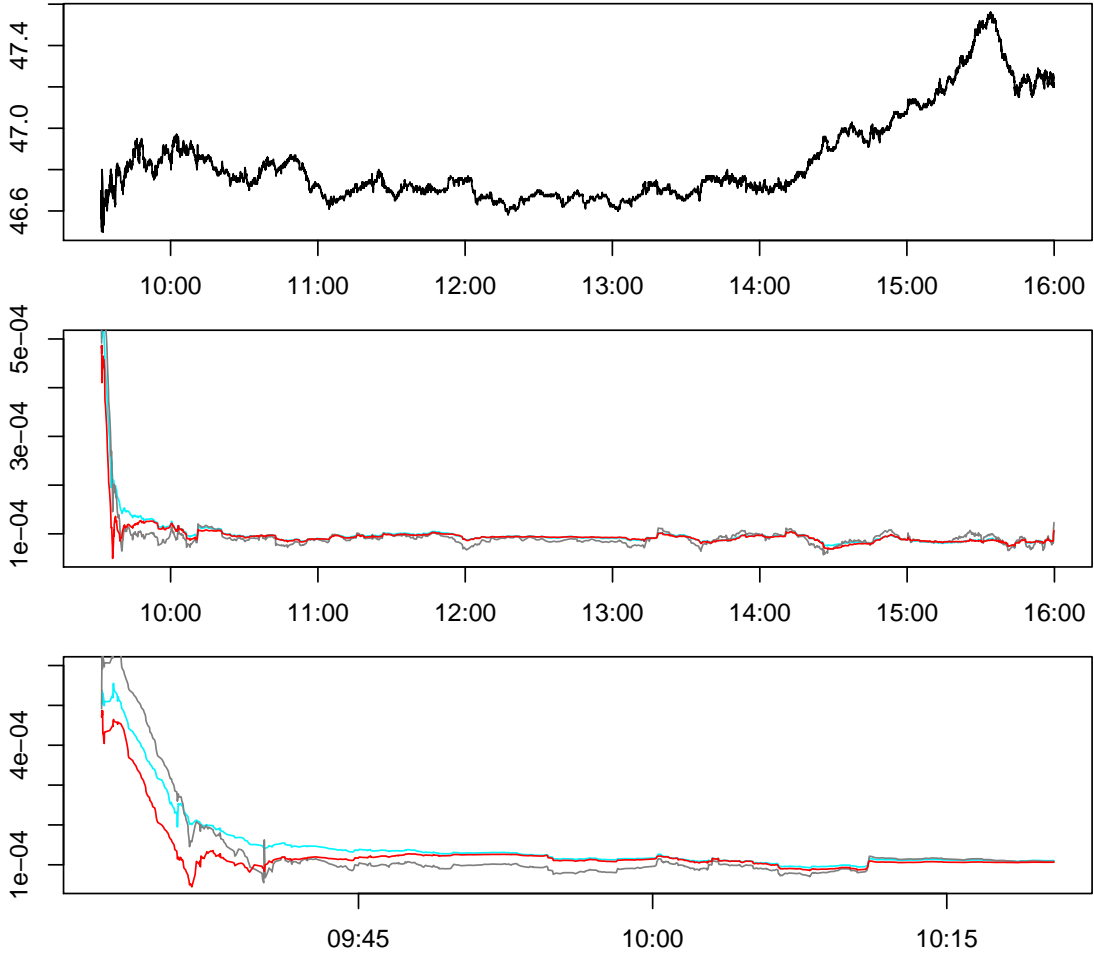


Figure 7: Real data example: Estimation of time-varying spot volatility in transaction time. The upper plot shows the transaction data of the symbol C for the 3rd September 2007. The middle and the lower plot give the volatility estimators  $\hat{\Sigma}_{t_j}^c$  (turquoise line),  $\hat{\Sigma}_{t_j|t_j}^*$  (red line) and the benchmark estimator  $\hat{\Sigma}_{t_j}^B$  (gray line).

### Clock time spot volatility estimation

We now compare our two approaches for the estimation of spot volatility in clock time for symbol C. The first estimator  $\hat{\Sigma}_{t_j}^{*c}$  is applied as described in Sections 4.1 and 5.2.  $\alpha$  and  $\beta$  are optimized with respect to (29) where  $\check{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}})$  is replaced with  $\check{\Sigma}_{t_{j+1}}^c(\omega_{t_{j+1}}^c)$ . A plot of this estimator (not presented) was quite poor – apart from some strong spikes caused by very small values of  $t_j - t_{j-1}$  and therefore very large values of  $\check{\Sigma}_{t_j}^c(\omega_{t_j}^c)$  in (25), the volatility seemed to be strongly oversmoothed. This is caused by the MSE-type criterion in (29) in combination with the very large values of  $\check{\Sigma}_{t_j}^c(\omega_{t_j}^c)$  acting like outliers and leading to small  $\lambda_j$ . We therefore intuitively took  $2\lambda_j$  leading to the estimate that is plotted in Figure 8. The second estimator is the alternative estimator  $\hat{\Sigma}_{\text{alt}}^c(t_j) = \hat{\Sigma}_{t_j|t_j}^*/\bar{\delta}_j$  proposed in Section 4.2 with the transaction time estimator  $\hat{\Sigma}_{t_j|t_j}^*$  from Figure 7 (red line). For the duration estimator  $\bar{\delta}_j$  we found empirically that a constant step size suffices (because the duration curve roughly has constant smoothness over the trading day). The used step size for  $\bar{\delta}_j$  is determined by minimizing the prediction error  $\sum_{j=2}^{T-1} \{\bar{\delta}_j - (t_{j+1} - t_j)\}^2$  leading to  $\lambda = 0.1025$ . (We mention that because of the dependence of the durations  $\bar{\delta}_j$  and  $(t_{j+1} - t_j)$  usually are not independent and the minimization of the above criterion therefore is

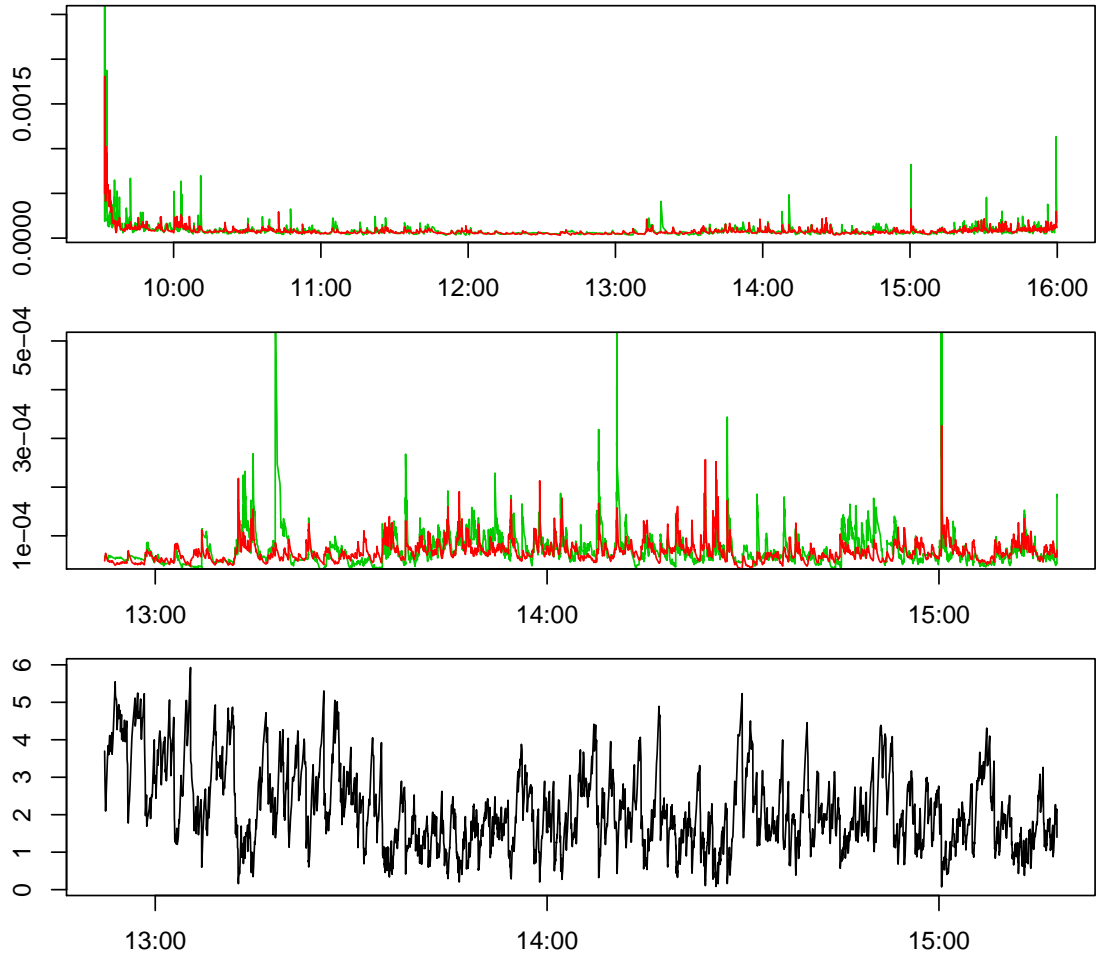


Figure 8: Real data example: Estimation of time-varying spot volatility in clock time based on the transactions of symbol C for the 3rd September 2007. The upper plot gives the volatility estimators  $\tilde{\Sigma}_{t_j|t_j}^{*c}$  (green line) and  $\hat{\Sigma}_{\text{alt}}^c(t_j)$  (red line). The middle plot shows the estimators for a fraction of the trading day. The averaged duration times  $\bar{\delta}_j$  (for a fraction of the trading day) are given in the lower plot (the y-axis shows seconds).

not approximately the same as the minimization of the mean squared error. Despite of this we think that the resulting  $\lambda$  is reasonable. However, this should be investigated further.)

The estimation results are provided in Figures 8 and 9. First we state that both estimators roughly coincide in magnitude (which was not clear beforehand). From the upper plot of Figure 8 we observe that  $\tilde{\Sigma}_{t_j|t_j}^{*c}$  (green line) produces some large spikes during the trading day (due to small values of  $t_j - t_{j-1}$ ). The variability of  $\hat{\Sigma}_{\text{alt}}^c(t_j) = \tilde{\Sigma}_{t_j|t_j}^{*c} / \bar{\delta}_j$  is mainly a result of the variability of the duration estimator  $\bar{\delta}_j$  (plotted in the lower plot) because the transaction time estimator  $\tilde{\Sigma}_{t_j|t_j}^{*c}$  is almost constant (apart from the beginning of the trading day – see Figure 7). The fluctuation of the duration estimator is very high during the whole day.

Figure 9 compares the transaction data and the volatility estimates for a small time period. The different behavior of the two estimators is apparent. We regard the strong spikes of  $\tilde{\Sigma}_{t_j|t_j}^{*c}$  as artificial due to small values of  $t_j - t_{j-1}$ . Furthermore, the estimator needs about one minute to settle down again after the occurrence of a spike. On the other hand the small spikes of  $\hat{\Sigma}_{\text{alt}}^c(t_j)$  are caused by small averaged durations. For this reason we have more confidence in the second estimator. In addition, it is theoretically more appealing (because the transaction time volatility is

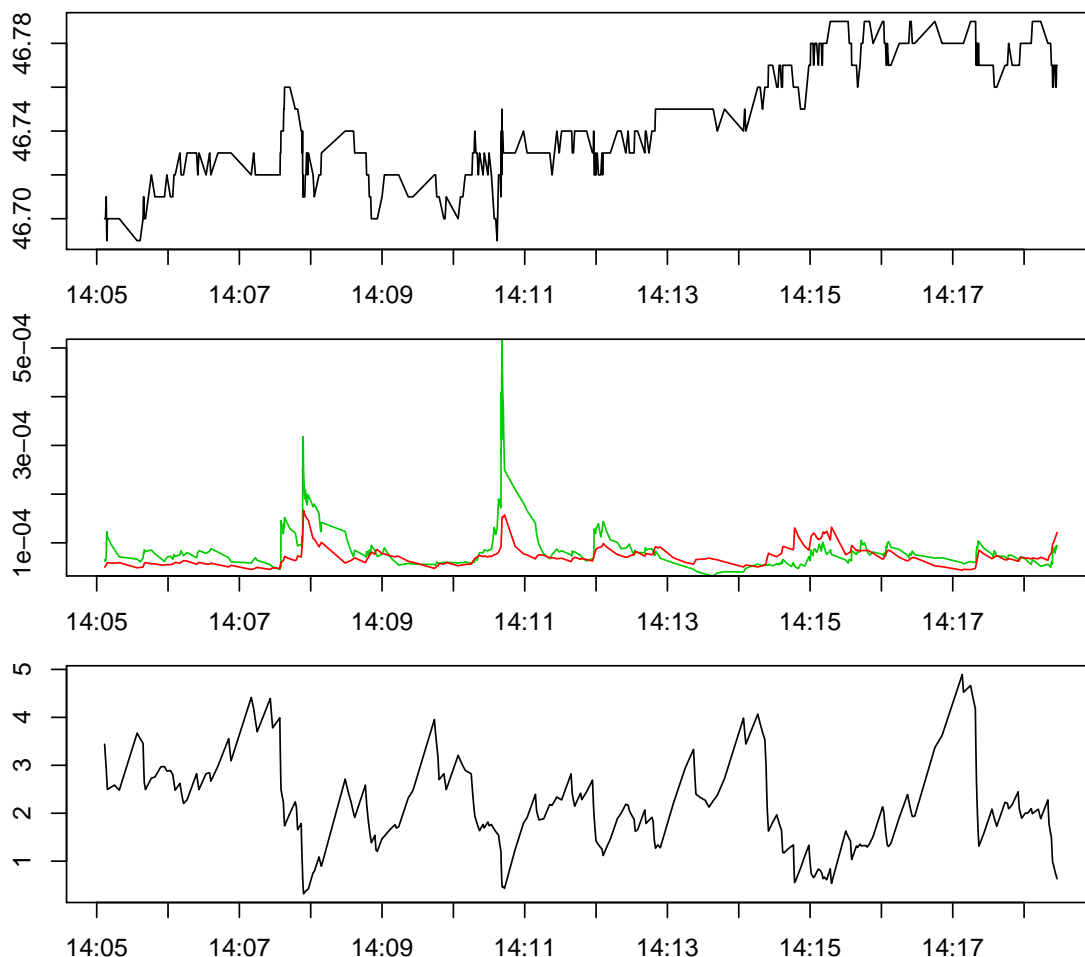


Figure 9: Real data example: Estimation of time-varying spot volatility in clock time based on the transactions of symbol C for the 3rd September 2007. The figure only gives the results for a small fraction of the trading day (compare Figure 8). The plots show (from top to bottom): transaction prices of C; our volatility estimators  $\hat{\Sigma}_{t_j|t_j}^{*c}$  (green line) and  $\hat{\Sigma}_{\text{alt}}^c(t_j)$  (red line); the averaged duration times  $\bar{\delta}_j$ .

almost constant and the variability of the clock time volatility is mainly caused by the variability of the trading intensity).

The second estimator is also more stable for another reason: Because volatility in transaction time is less varying the particle filter in transaction time is more stable.

## 8 Concluding Remarks

### Methodological Comments

We have presented a new technique for the on-line estimation of time-varying volatility based on noisy transaction data. Our algorithm is easy to implement and computationally efficient. It updates the volatility estimate immediately after the occurrence of a new transaction, and it therefore is as close to the market as possible. It also corrects for the bias which occurs as a result of the on-line estimation. It is straightforward to extend our method to more complicated price models (e.g. with a drift term) or other microstructure noise models.

Our work was guided by the goal to execute all calculations on-line in a high-frequency situation, and, at the same time, to base all methods on solid statistical principles. We feel that this goal has been achieved: Our algorithm is computationally efficient and it can be applied in real-time. On a recent personal computer an efficient implementation of our method requires a few milliseconds for a single update of the estimator (including one iteration of the particle filter with 500 particles). At the same time we use established or new statistical methods such as particle filters in nonlinear state space models, EM-type algorithms, and adaptation by quasi mean squared error minimization.

The contribution of this work is manifold. First, we have proposed a nonlinear market microstructure noise model that covers bid-ask bounces, time-varying bid-ask spreads, and the discreteness of prices observed in real data. Second, the problem of on-line volatility estimation has been treated in a nonlinear state-space framework. It has been shown that the filtering distribution of the efficient price can be approximated with a particle filter and that the volatility can be estimated as a parameter of the filtering distribution. Third, we have presented a new bias-corrected sequential EM-type algorithm which allows the on-line estimation of time-varying volatility. Fourth, the problem of on-line adaptation has been treated satisfactorily (although still a bit ad-hoc from a theoretical viewpoint). The usefulness of the approach for real-time applications has been demonstrated through Monte Carlo simulations and applications to stock data.

## **Practical Aspects**

Besides the new microstructure noise model we make a clear distinction between the (spot) volatility per time unit  $\Sigma^c(t)$  and the volatility per transaction  $\Sigma(t)$ . Volatility in clock time usually is much more volatile than volatility in transaction time. We advocate the use of transaction time for modeling, i.e. to estimate  $\Sigma(t)$ , together with a subsequent transformation based on the trading intensity to obtain an estimator for  $\Sigma^c(t)$ . At least for our data sets it turned out that volatility in transaction time is almost constant (apart from the beginning of the trading day) and the fluctuation of clock time volatility is merely a result of fluctuation of the trading intensity (or the mean duration between subsequent trades). Thus a new focus in volatility estimation may be on the modeling of trading times. It is an interesting open question whether major external events do not only cause an increase in trading intensity but also an increase in transaction time volatility.

Furthermore, we are convinced that the distribution of asset returns in a transaction time model can be modeled in most situations quite well by a Gaussian distribution and many “jumps” observed in security prices sampled on an equally spaced clock time grid are due to a drastically increased number of transactions at that time. Our view is based on the investigation of several data sets (not reported in this paper).

Another issue is the question for the correct goal in volatility estimation: We think that practitioners are more interested in a rapidly adapting (i.e. close to unbiased) and undersmoothed



estimator instead of an oversmoothed estimator. In that case minimizing the mean squared error would not be the optimal strategy. We have presented in this paper with the approximately unbiased  $\tilde{\Sigma}_{t_j|t_j}$  an estimator in this direction.

## Mathematical Challenges

Of course it is desirable to have a complete mathematical theory on the methods of this paper. However, we think that this is very hard to achieve. Here are a few comments in detail:

The results on the particle filter are mathematically exact given that the true volatility is known (i.e. with  $\hat{\Sigma}_{t_j}^{\text{pf}} = \Sigma_{t_j}$ ) including the results from Proposition 1 on the optimal proposal and the importance weights. In particular it determines correctly the conditional distribution of the efficient prices given the observations.

Even in the case of constant volatility and for the simplest estimator  $\hat{\Sigma}_{t_j}$  from (21) it seems to be very difficult to establish consistency and the asymptotic distribution. In the slightly simpler context of i.i.d.-observations convergence properties of recursive EM-type algorithms have been studied in Titterton (1984), Sato (2000), Wang and Zhao (2006), and Cappé and Moulines (2009) where also proofs of consistency and asymptotic normality are provided.

For strict mathematical results on local consistency or asymptotic normality some rescaling framework would be necessary. One approach could be to let the sampling frequency tend to infinity which would mean in the present setting of non-equally spaced observations that  $\sup_j \tau'(j) \rightarrow 0$  where  $\tau$  is defined as in Section 5.2. At the same time the maximal step size had to go to zero, i.e.  $\sup_j \lambda_j \rightarrow 0$ . Furthermore the assumption  $\sup_j \tau'(j) / \inf_j \lambda_j \rightarrow 0$  would be needed (this corresponds to the common assumption  $n \rightarrow \infty$ ,  $bn \rightarrow 0$  and  $b \rightarrow 0$  for kernel estimates with bandwidth  $b$ ). All “ $\approx$ ”-signs in the appendix and most of the “ $\approx$ ”-signs in Section 5.2 mean that the remaining terms are of lower order if these assumption were fulfilled.

An even bigger challenge is to determine the approximate mean squared error for the estimate (46) (with the particles instead of the efficient price as in the appendix). This would require to prove the “ $\approx$ ”-sign in (31) and (even harder) to prove the corresponding relation for the variance.

A strict mathematical result on bias reduction by combining two on-line algorithms with different step sizes (similar to (36) but with time-constant step sizes) has been proved in the context of time-varying ARCH models in Dahlhaus and Subba Rao (2007).

Finally, it is a mathematical challenge to put the interplay between transaction time volatility and clock time volatility on solid mathematical grounds - for example by proving consistency of the estimator  $\hat{\Sigma}_{\text{alt}}^c(t_j) = \tilde{\Sigma}_{t_j|t_j}^* / \bar{\delta}_j$  in a subordinated differential equation model  $d\mathbf{X}(t) = \Gamma(t) d\mathbf{W}_{N(t)}$  with an adequate point process  $N(t)$ .

## Outlook

In a forthcoming paper we extend the results to the on-line estimation of time-varying cross-volatilities for non-synchronous trading times. This leads to a non-standard state-space model

where the components of the state evolve non-synchronously in different discrete times. The properties of this non-standard state-space model differ significantly from those of standard state-space models. In particular, the state process does not fulfill a Markov property. We develop a new particle filter that can cope with this situation.

## 9 Appendix

**9.1 Proof of Proposition 1:** The likelihood  $p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})$  is equal to one if  $x_{t_j,s} \in \log A_{t_j,s}$  for all  $s = 1, \dots, S$  and zero otherwise, that is

$$p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) = \prod_{s=1}^S \mathbf{1}_{\{x_{t_j,s} \in \log A_{t_j,s}\}}.$$

This and (6) recursively imply the uniqueness of the conditional distribution  $p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}})$ . (Note that  $p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}})$  does not depend on  $\mathbf{x}_{t_{1:j}}$  and is therefore part of the norming constant.) It is easy to verify that the optimal proposal satisfies

$$p(\mathbf{x}_{t_j} | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}) \propto p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}).$$

Furthermore, the transition prior is given by  $p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}) = \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}; \Sigma_{t_j})$  leading to the assertion. The expression for the importance weights follows from

$$p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_{j-1}}^i) = \int p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i) d\mathbf{x}_{t_j} = \int_{\log \mathbf{A}_{t_j}} p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i) d\mathbf{x}_{t_j}.$$

**9.2 Calculation of the quasi mean squared error in Section 5.2:** We now calculate and minimize the mean squared error of

$$\tilde{\Sigma}_{t_i|t_j}(\lambda) := (1 + \kappa) \hat{\Sigma}_{t_j} - \kappa \hat{\Sigma}_{t_j}^{(1/2)}$$

as an estimator of  $\Sigma(t_i)$  with respect to  $\kappa$ . For several reasons the variance of the estimator is very hard to derive (because of the recursive estimation scheme and the nonlinear microstructure noise model). In order not to overstress heuristic considerations we minimize instead the mean squared error of the above estimator in the case where the unknown efficient prices are used instead of the filter particles and call this the quasi mean squared error.

We only give a brief sketch. As in Section 5 we only discuss the univariate case. We obtain as in (32)

$$\mathbf{E} \tilde{\Sigma}_{t_i|t_j}(\lambda) \approx \Sigma(t_i) - [(1 + \kappa)(i - \bar{j}) - \kappa(i - \bar{j}^{(1/2)})] \dot{\Sigma}(\bar{j}) \quad (46)$$

and for the variance

$$\begin{aligned} \text{Var}(\hat{\Sigma}_{t_j}) &\approx \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell})^2 \right] \lambda_{j-k}^2 2 \Sigma(t_{j-k})^2 + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell})^2 \right] 2 \Sigma(t_2)^2 \\ &\approx \left[ \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell})^2 \right] \lambda_{j-k}^2 + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell})^2 \right] \right] 2 \Sigma(\bar{t}_j)^2. \end{aligned} \quad (47)$$

Similarly we obtain

$$\text{Var}(\hat{\Sigma}_{t_j}^{(1/2)}) \approx \left[ \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} \left(1 - \frac{\lambda_{j-\ell}}{2}\right)^2 \right] \frac{\lambda_{j-k}^2}{4} + \left[ \prod_{\ell=0}^{j-3} \left(1 - \frac{\lambda_{j-\ell}}{2}\right)^2 \right] \right] 2 \Sigma(t_{\bar{j}})^2$$

and

$$\text{Cov}(\hat{\Sigma}_{t_j}, \hat{\Sigma}_{t_j}^{(1/2)}) \approx \left[ \sum_{k=0}^{j-3} \left[ \prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \left(1 - \frac{\lambda_{j-\ell}}{2}\right) \right] \frac{\lambda_{j-k}^2}{2} + \left[ \prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \left(1 - \frac{\lambda_{j-\ell}}{2}\right) \right] \right] 2 \Sigma(t_{\bar{j}})^2.$$

The terms in the brackets can be calculated by the recursions (37) through (39). Therefore

$$\begin{aligned} \text{Var}(\tilde{\Sigma}_{t_i|t_j}(\lambda)) &\approx \left[ (1 + \kappa)^2 v_{1,j} + \kappa^2 v_{2,j} - 2(1 + \kappa) \kappa v_{3,j} \right] 2 \Sigma(t_{\bar{j}})^2 \\ &= \left[ v_{1,j} + \kappa (2v_{1,j} - 2v_{3,j}) + \kappa^2 (v_{1,j} + v_{2,j} - 2v_{3,j}) \right] 2 \Sigma(t_{\bar{j}})^2 \end{aligned}$$

leading to the mean squared error

$$\begin{aligned} \mathbf{E} \left( \tilde{\Sigma}_{t_i|t_j}(\lambda) - \Sigma(t_i) \right)^2 &\approx \left[ - (1 + \kappa)(i - \bar{j}) + \kappa (i - \bar{j}^{(1/2)}) \right]^2 \dot{\Sigma}(\bar{j})^2 \\ &\quad + \left[ v_{1,j} + \kappa (2v_{1,j} - 2v_{3,j}) + \kappa^2 (v_{1,j} + v_{2,j} - 2v_{3,j}) \right] 2 \Sigma(t_{\bar{j}})^2. \end{aligned}$$

Minimization with respect to  $\kappa$  yields with  $\dot{\Sigma}(j) = \Sigma'(t_j) \tau'(j)$

$$\begin{aligned} \kappa_{\min} &= \frac{(i - \bar{j})(\bar{j} - \bar{j}^{(1/2)}) \dot{\Sigma}(\bar{j})^2 - 2(v_{1,j} - v_{3,j}) \Sigma(t_{\bar{j}})^2}{(\bar{j} - \bar{j}^{(1/2)})^2 \dot{\Sigma}(\bar{j})^2 + 2(v_{1,j} + v_{2,j} - 2v_{3,j}) \Sigma(t_{\bar{j}})^2} \\ &= \frac{(i - \bar{j})(\bar{j} - \bar{j}^{(1/2)}) \left[ \frac{\partial}{\partial t} \log \Sigma(t) \Big|_{t=t_{\bar{j}}} \tau'(\bar{j}) \right]^2 - 2(v_{1,j} - v_{3,j})}{(\bar{j} - \bar{j}^{(1/2)})^2 \left[ \frac{\partial}{\partial t} \log \Sigma(t) \Big|_{t=t_{\bar{j}}} \tau'(\bar{j}) \right]^2 + 2(v_{1,j} + v_{2,j} - 2v_{3,j})}. \end{aligned}$$

## References

- Aït-Sahalia, Y., Mykland, P.A., and Zhang, L. (2005) How Often to Sample a Continuous-Time Process in the Presence of Market Microstructure Noise. *Review of Financial Studies*, 18, 351-416.
- Andersen, T.G., Bollerslev, T., and Meddahi, N. (2006) Realized Volatility Forecasting and Market Microstructure Noise. unpublished manuscript.
- Ané, T., and Geman, H. (2000) Order Flow, Transaction Clock, and Normality of Asset Returns. *The Journal of Finance*, 55, 2259-2284.
- Ball, C.A. (1988) Estimation Bias Induced by Discrete Security Prices. *The Journal of Finance*, 43, 841-865.
- Bandi, F.M., and Russell, J.R. (2006) Separating microstructure noise from volatility. *Journal of Financial Economics*, 79, 655-692.
- (2008) Microstructure noise, realized variance, and optimal sampling. *Review of Economic Studies*, 75, 339-369.
- Barndorff-Nielsen, O.E., Hansen, P.R., Lunde, A., and Shephard, N. (2008) Designing Realized Kernels to Measure the Ex-Post Variation of Equity Prices in the Presence of Noise. *Econometrica*, 76, 1481-1536.
- Bos, C.S., Janus, P., and Koopman, S.J. (2009) Spot Variance Path Estimation and its Application to High Frequency Jump Testing. Discussion Paper TI 2009-110/4, Tinbergen Institute.

- Briers, M., Doucet, A., and Maskell, S. (2010) Smoothing Algorithms for State-Space Models. *Annals Institute Statistical Mathematics*, 62, 61-89.
- Cappé, O., and Moulines, E. (2009) On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society, Series B*, 71, 593-613.
- Christensen, K., Podolskij, M., and Vetter, M. (2009) Bias-correcting the realised range-based variance in the presence of market microstructure noise. *Finance and Stochastics*, 13, 239-268.
- Dahlhaus, R., and Subba Rao, S. (2007) A recursive online algorithm for the estimation of time-varying ARCH parameters. *Bernoulli*, 13, 389-422.
- Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39, 1-38.
- Douc, R., Cappé, O., and Moulines, E. (2005) Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 64-69.
- Doucet, A., Godsill, S., and Andrieu, C. (2000) On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10, 197-208.
- Doucet, A., de Freitas, N., and Gordon, N. (ed.) (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.
- Engle, R.F., and Russell, J.R. (1998) Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66, 1127-1162.
- Even-Dar, E., and Mansour, Y. (2003) Learning Rates for Q-learning. *Journal of Machine Learning Research*, 5, 1-25.
- Fan, J., and Wang, Y. (2008) Spot volatility estimation for high-frequency data. *Statistics and Its Interface*, 1, 279-288.
- Foster, D., and Nelson, D. (1996) Continuous Record Asymptotics for Rolling Sample Estimators. *Econometrica*, 64, 139-174.
- Gabaix, X., Gopikrishnan, P., Plerou, V., and Stanley, H.E. (2003) A theory of power-law distributions in financial market fluctuations. *Nature*, 423, 267-270.
- Godsill, S.J., Doucet, A., and West, M. (2004) Monte Carlo Smoothing for Nonlinear Time Series. *Journal of American Statistical Association*, 99, 156-168.
- Gordon, N., Salmond, D., and Smith, A. (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140, 107-113.
- Hansen, P.R., and Lunde, A. (2006) Realized Variance and Market Microstructure Noise. *Journal of Business and Economics Statistics*, 24, 127-161.
- Harris, L. (1990) Estimation of Stock Price Variances and Serial Covariances from Discrete Observations. *Journal of Financial and Quantitative Analysis*, 25, 291-306.
- Howison, S., and Lamper, D. (2001) Trading volume in models of financial derivatives. *Applied Mathematical Finance*, 8, 119-135.
- Jacod, J., Li, Y., Mykland, P.A., Podolskij, M., and Vetter, M. (2009) Microstructure noise in the continuous case: The pre-averaging approach. *Stochastic Processes and their Applications*, 119, 2249-2276.
- Kalnina, I., and Linton, O. (2008) Estimating quadratic variation consistently in the presence of endogenous and diurnal measurement error. *Journal of Econometrics*, 147, 47-59.
- Kong, A., Liu, J., and Wong, W. (1994) Sequential imputation and Bayesian missing data problems. *Journal of American Statistical Association*, 89, 278-288.
- Kristensen, D. (2009) Nonparametric Filtering of the Realised Spot Volatility: A Kernel-based Approach. *Econometric Theory*, in press.

- Large, J. (2007) Estimating Quadratic Variation When Quoted Prices Change By A Constant Increment. unpublished manuscript.
- Li, Y., and Mykland, P.A. (2007) Are volatility estimators robust with respect to modeling assumptions?. *Bernoulli*, 13, 601-622.
- Meddahi, N., Renault, E., and Werker, B. (2006) GARCH and irregularly spaced data. *Economic Letters*, 90, 200-204.
- Munk, A., and Schmidt-Hieber, J. (2009) Nonparametric Estimation of the Volatility Function in a High-Frequency Model corrupted by Noise. unpublished manuscript.
- Neddermeyer, J.C. (2010) Nonparametric Particle Filtering and Smoothing with Quasi-Monte Carlo Sampling. *Journal of Statistical Computation and Simulation*, in press.
- Plerou, V., Gopikrishnan, P., Gabaix, X., A Nunes Amaral, L., and Stanley, H.E. (2001) Price fluctuations, market activity and trading volume. *Quantitative Finance*, 1, 262-269.
- Podolskij, M., and Vetter, M. (2009) Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps. *Bernoulli*, 15, 634-658.
- Robert, C.Y., and Rosenbaum, M. (2008) Ultra high frequency volatility and co-volatility estimation in a microstructure model with uncertainty zones. unpublished manuscript.
- Rosenbaum, M. (2009) Integrated volatility and round-off error. *Bernoulli*, 15, 687-720.
- Sato, M. (2000) Convergence of on-line EM algorithm. In *Proc. Int. Conf. on Neural Information Processing*, 1, 476-481.
- Taylor, J.W. (2004) Volatility forecasting with smooth transition exponential smoothing. *International Journal of Forecasting*, 20, 273-286.
- Titterton, D.M. (1984) Recursive Parameter Estimation Using Incomplete Data. *Journal of the Royal Statistical Society, Series B*, 46, 257-267.
- Voev, V., and Lunde, A. (2007) Integrated Covariance Estimation using High-Frequency Data in the Presence of Noise. *Journal of Financial Econometrics*, 5, 68-104.
- Wang, S., and Zhao, Y. (2006) Almost sure convergence of Titterton's recursive estimator for mixture models. *Statistics & Probability Letters*, 76, 2001-2006.
- Zeng, Y. (2003) A Partially Observed Model for Micromovement of Asset Prices with Bayes Estimation via Filtering. *Mathematical Finance*, 13, 411-444.
- Zhang, L., Mykland, P.A., and Ait-Sahalia (2005) A Tale of Two Time Scales: Determining Integrated Volatility with Noisy High-Frequency Data. *Journal of the American Statistical Association*, 100, 1394-1411.
- Zhou, B. (1996) High-Frequency Data and Volatility in Foreign-Exchange Rates. *Journal of Business & Economic Statistics*, 14, 45-52.
- Zumbach, G, Corsi, F., and Trapletti, A. (2002) Efficient estimation of volatility using high-frequency data. Technical Report, Olsen & Associates.