How Frequently Does the Stock Price Jump? – An Analysis of High-Frequency Data with Microstructure Noises

Jin-Chuan Duan and Andras Fulop[*]

(First Draft: September 15, 2006)
(This Draft: February 28, 2007)

Abstract

The stock price is assumed to follow a jump-diffusion process which may exhibit time-varying volatilities. An econometric technique is then developed for this model and applied to high-frequency time series of stock prices that are subject to microstructure noises. Our method is based on first devising a localized particle filter and then employing fixed-lag smoothing in the Monte Carlo EM algorithm to perform the maximum likelihood estimation and inference. Using the intra-day IBM stock prices, we find that high-frequency data are crucial to disentangling frequent small jumps from infrequent large jumps. During the trading sessions, jumps are found to be frequent but small in magnitude, which is in sharp contrast to infrequent but large jumps when the market is closed. We also find that at the 5- or 10-minute sampling frequency, the conclusion will critically depend on whether heavy-tailed microstructure noises have been accounted for. Ignoring microstructure noises can, for example, lead to an overestimation of the jump intensity of 50% or more.

Keywords: Particle filtering, jump-diffusion, maximum likelihood, EM-algorithm

JEL classification code: C22

[*] Duan is with Joseph L. Rotman School of Management, University of Toronto and Fulop is with ESSEC Paris and CREST. Duan acknowledges support received as the Manulife Chair in Financial Services and research funding from both the Social Sciences and Humanities Research Council of Canada and the Natural Sciences and Engineering Research Council of Canada. Duan also thanks the supports received during his visit to Guanghua School of Management, Peking University where the paper was completed. E-mail: jcduan@rotman.utoronto.ca. Fulop would like to acknowledge the hospitality of the National Bank of Hungary where parts of this research were completed. E-mail: fulop@essec.fr. This paper has been presented at Chinese Academy of Sciences (Institute of Applied Mathematics), National Seoul University (Department of Economics), Chinese University of Hong Kong (Department of Statistics), National University of Singapore (Risk Management Institute), CREST, National Bank of Hungary, AFFI 2006. The authors acknowledge the comments received in the above seminars/workshops.

1 Introduction

Arguably, few financial researchers or market practitioners will question the premise that stock prices face jumps. The exact nature of jumps is, however, subject to much debate. Do stock prices jump frequently? What are the jump magnitudes? Major market news are often released, by design, after the trading session to allow for an orderly digestion of information. Shouldn't this practice imply that jumps are likely to be infrequent but large in magnitude when the market is closed vis-a-vis open? Are the observed jumps a result of microstructure noises or due to fundamental changes in the "efficient" stock value?

Intuitively, high-frequency data will be critical to answering the aforementioned questions. With the availability of intra-day stock prices, one can begin to address this issue in a more intelligent way. In a similar spirit, recent advancements on "realized volatility" based on high-frequency data are, for example, made possible by the availability of intra-day data. However, using high-frequency data can have its hazards.[1] The observed prices are contaminated by microstructure noises naturally arising from trading based on information and/or liquidity. The observed prices are also subject to tick-size discretization. While these may not have much effect when one deals with daily or lower-frequency data, microstructure noises are likely to have increased importance when one moves to high-frequency data.

This paper devises an econometric technique attempting to shed light on the nature of stock price jumps. Specifically, we assume the stock price follows a jump-diffusion process with the volatility being time-varying. Time-varying volatility is captured via using the so-called realized volatility computed from high-frequency data. Microstructure noises of two types are incorporated into our model. First, the information/liquidity induced microstructure noises are proxied by a heavy-tailed distributed measurement error. Second, the tick-size induced distortion is explicitly accounted for in our model. The resulting specification is a highly complex nonlinear state-space model with non-Gaussian random variables. Our solution technique relies on particle filtering (a sequential Monte Carlo technique), a recent advancement for solving non-linear, non-Gaussian filtering problems.

In a parametric context, there are a number of papers estimating models with jumps using daily stock prices (sometimes supplemented with derivative prices); for example, Eraker, *et al* (2003), Bates (2000), and Pan (2002). These papers tend to find less frequent but larger jumps vis-a-vis the results obtained in this paper. This is not at all surprising because frequent small jumps in the stock value may give rise to an appearance of infrequent large jumps if one only uses lower-frequency data such as daily. Building on the realized volatility literature, Tauchen and Zhou (2005) use daily measures of quadratic and bipolar variations to test whether there were jumps on a given day. They then use the days with jumps to estimate the jump intensity and magnitude. A crucial identifying assumption is that on a given day there is at most one jump and the price

[1] Ait-Sahalia, *et al* (2005a) and Bandi and Russell (2006), for example, show that microstructure noises can induce a bias in the realized volatility estimate using high-frequency data.

movement on that day is due to the jump. Thus, Tauchen and Zhou's (2005) approach is really about pinning down infrequent large jumps. Our parametric approach is thus complementary to their nonparametric approach and allows for identifying frequent small jumps. This paper can also be viewed as a generalization to the parametric approach of Ait-Sahalia, *et al* (2005b) to take full advantage of the information contained in high-frequency data.

This paper offers two econometric innovations. First, we design a new particle filter to deal with jumps in high-frequency data. A tailor-made particle filter is needed because jumps in high-frequency setting inevitably lead to extremely peaked densities making Monte Carlo approximations poor. Our device is to localize the particle filter using four subsets of particles with each corresponding to one of the four possible combinations of jump/no jump in the stock price and the measurement error for each forward time step. Doing so ensures that an occurrence of jump in a tiny time step, albeit unlikely from a probabilistic point of view, will always be considered. If an actual stock price points to a high likelihood of a jump at that moment, the updated filtering weights assigned to the corresponding subset of particles will become non-negligible, simply because a small ex-ante probability of the jump occurrence is offset by a high likelihood (peaked density) conditional on the actual stock price.

The second innovation hinges upon the recognition of the fact that resampling, a critical step for any particle filter, cannot be performed smoothly for our model even with the smoothed empirical filtering distribution, an innovative technique proposed by Pitt (2002).[2] In other words, the sample likelihood function obtained via the particle filter cannot be made smooth enough in relation to the model parameters, making it impossible to use gradient-based optimization and/or to conduct maximum likelihood inference. Our solution is to use the Monte Carlo EM algorithm to indirectly optimize the sample likelihood function where our localized particle filter plays a key role in performing efficiently the simulated E-step. In essence, we take advantage of the fact that the Monte Carlo average of the complete-data log-likelihood is a smooth function of the model parameters (the ones to be updated) even though it is not smooth with respect to the model parameters being used to compute the Monte Carlo average. Thus, the irregularity induced by the use of a non-smooth particle filter has been circumvented.

We use the intra-day stock price for IBM in our empirical implementation. Performing the estimation for data in 2004, several interesting findings are obtained. In particular, as one increases the sampling frequency from once every hour to once every 10 minutes, the estimated mean number of jumps in prices per trading session rises from 5.4 to 13.6. The jump size also depends on the sampling frequency. The standard deviation of the jump size drops from 0.3% to 0.17%. This finding suggests that frequent small jumps may be disguised as large infrequent jumps if the sampling frequency is low, a result that is intuitively plausible. Our second finding is that at the 5-minute or 10-minute sampling frequency, the jump intensity can be overestimated by more than 50% if one

[2] In a recent paper by Duan and Fulop (2006), Pitt's (2002) smoothing technique was successfully employed to deal with the structural credit risk model. We believe two factors contribute to its successful application there. First, the model has no jumps. Second, using daily data makes the empirical filtering distribution more regular.

ignores microstructure noises. At the hourly or 30-minute sampling frequency, however, ignoring microstructure noises does not seem to change the estimate for the jump intensity if one ignores microstructure noises. This is in line with the intuition that as the sampling frequency increases, microstructure noises become increasingly consequential. Third, our finding suggests that it is important to allow for heavy-tailed microstructure noises, perhaps to account for occasional large information-motivated deviations from the "efficient" price. Finally, we find a large difference in the nature of jumps when the market was open vis-a-vis closed. Jumps in the closed period seem to be infrequent but larger in magnitude, likely reflective of a common practice of announcing significant corporate news after the market is closed. To see whether the conclusions reached for the 2004 data hold true for other years, we repeat the estimation for 2002, 2003 and 2005. The results are consistent with those of 2004.

2 Estimation technique for a model of high-frequency stock prices

2.1 The model

The true logarithmic stock value process is assumed to follow a jump-diffusion process:

$$dX_t = (\mu_x - \rho X_t)dt + \sigma_{x,t} dW_t + J_t dN_t \tag{1}$$

where W_t is a Wiener process, N_t is a Poisson process with intensity $\lambda_{x,t}$, and J_t is a normally distributed jump size with mean μ_{J_x} and variance $\sigma^2_{J_x,t}$ which is independent of W_t and N_t. The Poisson intensity and jump size variance are allowed to depend on whether the market is open; that is,

$$\begin{aligned} \lambda_{x,t} &= \lambda_{x,op} 1_{\{t \in \mathcal{T}_{op}\}} + \lambda_{x,cl}(1 - 1_{\{t \in \mathcal{T}_{op}\}}) \\ \sigma_{J_x,t} &= \sigma_{J_x,op} 1_{\{t \in \mathcal{T}_{op}\}} + \sigma_{J_x,cl}(1 - 1_{\{t \in \mathcal{T}_{op}\}}) \end{aligned}$$

where $\mathcal{T}_{op}$ denotes the time set when the market is open. Note that we have allowed the process to be potentially mean-reverting, i.e., $\rho \neq 0$. For our implementation later, we set $\rho = 0$ to reflect the typical random walk assumption for stock prices. We state the problem in a more general setup so that the method can be applied to other financial time series such as interest rates that are expected to exhibit a mean-reverting behavior. The local volatility $\sigma_{x,t}$ is allowed to be time-varying. Its exact specification will be described later.

The observed logarithmic stock prices are assumed to be contaminated by microstructure noises of two types - trading effects due to illiquidity and asymmetric information being the first and tick size the second. We assume that the first type of microstructure noise is composed of a normally distributed term plus a Bernoulli event with a normally distributed magnitude. Specifically, the contaminated logarithmic stock price before subjecting to the tick size adjustment is

$$Y_t = X_t + \varepsilon_t + q_t \xi_t \tag{2}$$

where ε_t and ξ_t are independent normal random variables with zero means and variances σ_y and σ_{J_y}, and q_t is a Bernoulli random variable independent of ε_t and ξ_t. λ_y is the probability for $q_t = 1$ and $1 - \lambda_y$ is for $q_t = 0$.

For lower-frequency data, it is both customary and reasonable to ignore the effect of tick size. For high-frequency data, however, tick size needs to be explicitly incorporated so as not to bias a study's conclusion. The presence of tick size complicates the matter significantly. We assume that the contaminated stock price is rounded to the nearest tick with the tick size being c, which means that the observed stock price is

$$S_t = c\left[\frac{\exp(Y_t)}{c} + 0.5\right] \tag{3}$$

where $[\cdot]$ denotes an integer operator which takes the value inside down to the nearest integer. The NYSE and other US stock exchanges were ordered to switch to the decimal pricing by April 9, 2001. The switch-over at the NYSE actually began on August 28, 2000 with seven stocks being traded in decimals in a pilot program. Later the decimal pricing was expanded to all stocks on January 29, 2001. Before the switch-over shares were traded in the NYSE in the multiples of one-sixteenth or one-eighth of a dollar, depending on the price range of a particular stock. This means that prior to the switch-over, $c = 0.0625$ or 0.125, and afterwards $c = 0.01$.

We denote the data set of observed stock prices by $\mathcal{D}_n \equiv \{s_0, s_1, s_2, \cdots, s_n\}$ which was sampled at time $\{t_0, t_1, t_2, \cdots, t_n\}$. Note that the sample need not be equally-spaced in time. To simplify notations, we denote the time between two sampling points by $\Delta t_i = t_i - t_{i-1}$, and use S_i to represent S_{t_i} and so on. The estimation task can then be formulated as the following non-linear, non-Gaussian state-space problem:

$$\begin{aligned} S_i &= c\left[\frac{\exp(Y_i)}{c} + 0.5\right] && (4)\\ Y_i &= X_i + \varepsilon_i + q_i\xi_i && (5)\\ X_i &= (1 - \rho\Delta t_i)X_{i-1} + \mu_x\Delta t_i + \sigma_{x,t_i}\Delta W_i + J_i\Delta N_i && (6) \end{aligned}$$

Note that equation (6) is based on an Euler approximation to equation (1) and ΔN_i is a Poisson random variable with $\lambda_{x,t_i}\Delta t_i$ as its parameter. The local volatility process σ_{x,t_i} is assumed to be measurable with respect to the information set generated by the observed stock prices up to t_{i-1}, i.e., $\mathcal{D}_{i-1}$. Under this assumption, the local volatility is in effect observable, and therefore need not be filtered which simplifies the estimation task. In our implementation later, we make local volatility dependent on the realized volatility computed from high-frequency data.

The non-linear, non-Gaussian filtering system is complex in two aspects. First, jumps in both the "efficient" stock value and the measurement error make the system non-Gaussian. Second, the tick size adjustment is a non-linear operation. Either fact renders the standard Kalman filtering technique or the extended Kalman filters unsuitable for the task in hand. It turns out that the standard particle filtering technique is also ill-suited for the problem because microstructure noises are typically small in magnitude, which means that the measurement equation is associated with a peaked density function. The problem is further complicated by the fact that high-frequency data by definition will associate the transition equation with an extremely peaked multi-modal density function. Interestingly, the peaked density problem can be resolved by suitably designing a local sampling-resampling scheme. Our proposed filtering algorithm is described next.

2.2 A localized particle filter

Our algorithm is based on the following decomposition of the joint filtering density/distribution:

$$\begin{aligned} & f\left(Y_i, X_i, X_{i-1}, q_i, \Delta N_i \middle| \mathcal{D}_i\right) \\ = \; & f\left(X_i \middle| Y_i, X_{i-1}, q_i, \Delta N_i\right) f\left(Y_i \middle| s_i, X_{i-1}, q_i, \Delta N_i\right) f\left(X_{i-1}, q_i, \Delta N_i \middle| \mathcal{D}_i\right) \\ \propto \; & f\left(X_i \middle| Y_i, X_{i-1}, q_i, \Delta N_i\right) f\left(Y_i \middle| s_i, X_{i-1}, q_i, \Delta N_i\right) \times \\ & f\left(s_i \middle| X_{i-1}, q_i, \Delta N_i\right) p(q_i, \Delta N_i) f\left(X_{i-1} \middle| \mathcal{D}_{i-1}\right) \end{aligned} \tag{7}$$

The last expression in (7) suggests a way to sample from the filtering distribution, given a sample of particles representing $f\left(X_{i-1} \middle| \mathcal{D}_{i-1}\right)$. First, augment the old particles with jumps, i.e., extending the state-space to include jumps in the system. Then, perform resampling to obtain the particle $(X_{i-1}, q_i, \Delta N_i)$ based on the weights $f\left(s_i \middle| X_{i-1}, q_i, \Delta N_i\right) p(q_i, \Delta N_i)$. This step amounts to “peaking into the future” because resampling has yielded a sample that uses the knowledge of s_i. This approach is analogous to the idea of auxiliary particle filtering in Pitt and Shephard (1999). Finally, sample (X_i, Y_i) according to $f\left(X_i \middle| Y_i, X_{i-1}, q_i, \Delta N_i\right) f\left(Y_i \middle| s_i, X_{i-1}, q_i, \Delta N_i\right)$.

The necessary quantities for executing this algorithm are described below. For the joint jump probability, we have

$$p(q_i, \Delta N_i) = p(q_i) p(\Delta N_i) \tag{8}$$

where

$$p(q_i = 0) = 1 - \lambda_y \text{ and } p(q_i = 1) = \lambda_y \tag{9}$$

$$p(\Delta N_i = k) = \frac{(\lambda_{x,t_i} \Delta t_i)^k}{k!} e^{-\lambda_{x,t_i} \Delta t_i} \text{ for } k = 0, 1, 2, \cdots \tag{10}$$

We now turn to the expression for the conditional likelihood of the observed value, $f\left(s_i \middle| X_{i-1}, q_i, \Delta N_i\right)$. Equations (5) and (6) imply that the conditional distribution $f\left(Y_i \middle| X_{i-1}, q_i, \Delta N_i\right)$ is normal with mean and variance:

$$E\left[Y_i \middle| X_{i-1}, q_i, \Delta N_i\right] = (1 - \rho \Delta t_i) X_{i-1} + \mu_x \Delta t_i + \mu_{J_x} \Delta N_i \tag{11}$$

$$Var\left[Y_i \middle| X_{i-1}, q_i, \Delta N_i\right] = \sigma^2_{x,t_i} \Delta t_i + \sigma^2_{J_x,t_i} \Delta N_i + \sigma^2_y + 1_{\{q_i=1\}} \sigma^2_{J_y} \tag{12}$$

Corresponding to s_i, it must be that $Y_i \in \left[\ln(s_i - \frac{c}{2}), \ln(s_i + \frac{c}{2})\right)$. Thus, we can compute:

$$\begin{aligned} & f\left(s_i \middle| X_{i-1}, q_i, \Delta N_i\right) \\ = \; & \Phi\left(\frac{\ln(s_i + \frac{c}{2}) - E\left[Y_i \middle| X_{i-1}, q_i, \Delta N_i\right]}{\sqrt{Var\left[Y_i \middle| X_{i-1}, q_i, \Delta N_i\right]}}\right) - \Phi\left(\frac{\ln(s_i - \frac{c}{2}) - E\left[Y_i \middle| X_{i-1}, q_i, \Delta N_i\right]}{\sqrt{Var\left[Y_i \middle| X_{i-1}, q_i, \Delta N_i\right]}}\right) \end{aligned} \tag{13}$$

where $\Phi(\cdot)$ stands for the standard normal cumulative distribution function.

For $f\left(Y_i|\, s_i, X_{i-1}, q_i, \Delta N_i\right)$, one only needs to recognize that it is identical to $f\left(Y_i|\, X_{i-1}, q_i, \Delta N_i\right)$ truncated to the interval $\left[\ln(s_i - \frac{c}{2}), \ln(s_i + \frac{c}{2})\right)$.

Finally, $f\left(X_i|\, Y_i, X_{i-1}, q_i, \Delta N_i\right)$ is a normal density function because by equations (5) and (6), we have

$$\begin{aligned} f(Y_i|\, X_i, q_i) &= \frac{1}{\sqrt{2\pi} v_y(q_i)} \exp\left\{-\frac{(Y_i - X_i)^2}{2v_y^2(q_i)}\right\} \\ &= \frac{1}{\sqrt{2\pi} v_y(q_i)} \exp\left\{-\frac{(X_i - Y_i)^2}{2v_y^2(q_i)}\right\} \qquad (14) \\ f(X_i|\, X_{i-1}, \Delta N_i) &= \frac{1}{\sqrt{2\pi} v_x(\Delta N_i)} \exp\left\{-\frac{(X_i - u_x(X_{i-1}, \Delta N_i))^2}{2v_x^2(\Delta N_i)}\right\} \qquad (15) \end{aligned}$$

where

$$\begin{aligned} v_y^2(q_i) &= \sigma_y^2 + 1_{\{q_i=1\}}\sigma_{J_y}^2 \qquad (16) \\ u_x(X_{i-1}, \Delta N_i) &= (1 - \rho\Delta t_i)X_{i-1} + \mu_x \Delta t_i + \mu_{J_x} \Delta N_i \qquad (17) \\ v_x^2(\Delta N_i) &= \sigma_{x,t_i}^2 \Delta t_i + \sigma_{J_x,t_i}^2 \Delta N_i \qquad (18) \end{aligned}$$

Thus,

$$\begin{aligned} & f(X_i|\, Y_i, X_{i-1}, q_i, \Delta N_i) \\ \propto\ & \frac{1}{2\pi v_y(q_i) v_x(\Delta N_i)} \exp\left\{-\frac{(X_i - Y_i)^2}{2v_y^2(q_i)} - \frac{(X_i - u_x(X_{i-1}, \Delta N_i))^2}{2v_x^2(\Delta N_i)}\right\} \\ =\ & \frac{1}{2\pi v_y(q_i) v_x(\Delta N_i)} \exp\left\{-\frac{\left(X_i - \frac{v_x^2(\Delta N_i)Y_i + v_y^2(q_i)u_x(X_{i-1}, \Delta N_i)}{v_y^2(q_i) + v_x^2(\Delta N_i)}\right)^2}{2\frac{v_y^2(q_i)v_x^2(\Delta N_i)}{v_y^2(q_i) + v_x^2(\Delta N_i)}}\right\} \qquad (19) \end{aligned}$$

This in turn implies that X_i has a conditional normal distribution with

$$\begin{aligned} E\left[X_i|\, Y_i, X_{i-1}, q_i, \Delta N_i\right] &= \frac{v_x^2(\Delta N_i)Y_i + v_y^2(q_i)u_x(X_{i-1}, \Delta N_i)}{v_y^2(q_i) + v_x^2(\Delta N_i)} \qquad (20) \\ Var\left[X_i|\, Y_i, X_{i-1}, q_i, \Delta N_i\right] &= \frac{v_y^2(q_i)v_x^2(\Delta N_i)}{v_y^2(q_i) + v_x^2(\Delta N_i)} \qquad (21) \end{aligned}$$

Our localized particle filter with M particles consists of the following steps:

- **Step 1:** Initializing the particle filter by sampling M times of ε_0, q_0 and ξ_0 according to equation (5), and then compute $x_0^{(m)} = \ln s_0 - \varepsilon_0^{(m)} - q_0^{(m)}\xi_0^{(m)}$ for $m = 1, 2, \cdots, M$. (The tick size effect is ignored in initializing the filter.)

- **Step 2:** For any t_i $(i = 1, 2, \cdots, n)$ and corresponding to each $x_{i-1}^{(m)}$, generate a set of four particles for $(q_i^{(m)}, \Delta N_i^{(m)})$. They are

$$\begin{aligned}
q_i^{(m)} &= 0 \qquad \Delta N_i^{(m)} = 0, \\
q_i^{(m)} &= 0 \qquad \Delta N_i^{(m)} \sim p(\Delta N_i \mid \Delta N_i > 0), \\
q_i^{(m)} &= 1 \qquad \Delta N_i^{(m)} = 0, \\
q_i^{(m)} &= 1 \qquad \Delta N_i^{(m)} \sim p(\Delta N_i \mid \Delta N_i > 0).
\end{aligned}$$

Note that $\Delta N_i^{(m)}$ in two cases are zero, but in the other two cases, it can be any value from the set $\{1, 2, \cdots\}$, which is sampled according to the conditional probability $p(\Delta N_i \mid \Delta N_i > 0)$. To arrive at an empirical representation of $f\left(X_{i-1}, q_i, \Delta N_i \middle| \mathcal{D}_i\right)$, we attach to $(x_{i-1}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$, each of the $4 \times M$ particles, an importance weight:

$$w_i^{(m;j,k)} = f\left(s_i \middle| x_{i-1}^{(m)}, q_i^{(m)} = j, \Delta N_i^{(m)} = k\right) p(q_i = j) \left[1_{\{k \geq 1\}} p(\Delta N_i > 0) + 1_{\{k=0\}} p(\Delta N_i = 0)\right].$$

The elements of the above expression are available in equations (9), (10) and (13). Let $\bar{w}_i^{(j,k)} = \frac{1}{M} \sum_{m=1}^{M} w_i^{(m;j,k)}$ for $j = 0, 1$ and $k = 0$ or $k \geq 1$. The likelihood value for the i-th observed stock price is the sum of four values corresponding to four subsets of particles:

$$\mathcal{L}_i = \bar{w}_i^{(0,0)} + \bar{w}_i^{(0,k\geq 1)} + \bar{w}_i^{(1,0)} + \bar{w}_i^{(1,k\geq 1)}$$

and the filtered jump/no jump probabilities are

$$\begin{aligned}
\text{Prob}\left\{ q_i = j, \Delta N_i = 0 \middle| s_0, s_1, \cdots, s_i \right\} &= \frac{\bar{w}_i^{(j,0)}}{\mathcal{L}_i} \text{ for } j = 0, 1 \\
\text{Prob}\left\{ q_i = j, \Delta N_i \geq 1 \middle| s_0, s_1, \cdots, s_i \right\} &= \frac{\bar{w}_i^{(j,k\geq 1)}}{\mathcal{L}_i} \text{ for } j = 0, 1.
\end{aligned}$$

- **Step 3:** Resample from the $4 \times M$ the particle set according to the probability $\pi_i^{(m;j,k)} = \frac{w_i^{(m;j,k)}}{M\mathcal{L}_i}$ to yield M equal-weight particles denoted by $(x_{i-1|i}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$. This equal-weight M-particle set is again an empirical representation of $f\left(X_{i-1}, q_i, \Delta N_i \middle| \mathcal{D}_i\right)$.

- **Step 4:** Corresponding to each particle $(x_{i-1|i}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$, sample from the truncated normal density $f(Y_i | s_i, x_{i-1|i}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$ to generate the particle $(y_i^{(m)}, x_{i-1|i}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$, which empirically represents $f(Y_i, X_{i-1}, q_i, \Delta N_i | \mathcal{D}_i)$.

- **Step 5:** Equations (20) and (21) make sampling from $f(X_i | y_i^{(m)}, x_{i-1|i}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$ a straightforward task. This yields M particles: $(x_i^{(m)}, y_i^{(m)}, x_{i-1|i}^{(m)}, q_i^{(m)}, \Delta N_i^{(m)})$, which represent $f\left(X_i, Y_i, X_{i-1}, q_i, \Delta N_i \middle| \mathcal{D}_i\right)$. One can then proceed to marginalize X_i (i.e., keeping only $x_i^{(m)}$) to have M particles (equal-weight) to represent the filtering distribution of X_i.

Remarks: It may appear more natural to directly sample M particles in Step 2 as opposed to $4M$ particles. This sample can be easily obtained by first sampling q_i and ΔN_i and then proceeding to sample X_i. (The importance weight will of course need to be adjusted accordingly.) However, such a sampling scheme would yield a poor particle filter mainly because the event of $\Delta N_i \geq 1$ can have an extremely small probability for high-frequency data. In other words, the simulated sample is likely to miss out the particles associated with jumps in the "efficient" stock value. When the data have been subjected to jumps, the simulated particle set will fail to include those points of extremely high likelihood.

Note that the particle filter provides a sample on the entire past of the system up to t_i. Any quantity of interest based on the past particles can be computed and carried forward alongside with X_i. This is true because at any time t_i, X_i is sufficient for moving the algorithm forward. Denote by I_i the quantity whose distribution is of interest; for example, one may be interested in $I_i = (X_0 + X_1 + \cdots + X_i)/(i+1)$. Then, in all of the preceding derivations one can use the vector (I_i, X_i) in place of X_i. Conditional on X_i, the system's forward evolution has nothing to do with I_i, and thus the algorithm remains unchanged. However, the output of the filter at any time t_i will be a set of particles representing the joint filtering distribution of (I_i, X_i), i.e., $f\,(I_i, X_i, |\, \mathcal{D}_i)$.

2.3 Monte Carlo EM algorithm

We now address the issue of computing the maximum likelihood (ML) estimates for the model parameters. The particle filtering algorithm described in the preceding section can generate the log-likelihood function for any fixed parameter values. However, it is ill-suited for finding the ML estimates because the log-likelihood function is inherently irregular with respect to the parameters even with the use of common random numbers. This irregularity arises from the resampling step required for any particle filter. It turns out that smooth resampling proposed by Pitt (2002) is still not smooth enough for the problem in hand, because the jump model in conjunction with high-frequency data inevitably makes the density function associated with the jump components extremely peaked and multi-modal. We thus adopt an indirect approach to the ML estimation via the EM algorithm of Dempster, *et al* (1977).

The EM algorithm is an alternative way of obtaining the ML estimate for the incomplete data model, where incomplete data refers to the situation that the model contains some random variable(s) without corresponding observations. In our case, the presence of microstructure noises makes the observed data generically incomplete; that is, one can think of the complete data as including both the true and observed stock prices.[3] The EM algorithm involves two steps - expectation and maximization – and hence its name. One first writes down the complete-data log-likelihood function. Since it is not observable, one needs to compute its expected value by conditioning on the

[3] In fact for the numerical efficiency reason, it will be better to also include the jumps in the complete data representation.

observed data in conjunction with some assumed parameter values. This completes the expectation step. In the maximization step, one finds the new parameter values that maximize the expected complete-data log-likelihood function. The updated parameter values are then used to repeat the E- and M-step until convergence. Interestingly, the EM algorithm will converge to the ML estimate under some regularity conditions.

For our ML estimation, the E-step due to its complexity will have to be computed using the particle filter, which means that we are using the Monte Carlo EM (MCEM) algorithm.[4] Casting optimization as an EM algorithm problem effectively circumvents the irregularity induced by the particle filter, because the E-step ensures that the expected complete-data log-likelihood function is smooth with respect to the model parameters that define the complete-data log-likelihood function. Even though the function is still inherently irregular in relation to the assumed parameter values used in computing the expectation, it becomes immaterial as far as optimization is concerned. In effect, one has decoupled optimization from filtering in each iteration.

In general, the complete-data representation of the model is not unique. The choice of representation can be crucial to the convergence speed of the EM algorithm. We define the complete data as $\{(Y_i, U_i, q_i, \Delta N_i); i = 0, ..., n\}$ where

$$U_i = (1 - q_i)\frac{\varepsilon_i}{\sigma_y} + q_i \frac{\varepsilon_i + \xi_i}{\sqrt{\sigma_y^2 + \sigma_{J_y}^2}} \tag{22}$$

Working with Y_i instead of the discretized observations s_i and including the jumps, q_i and ΔN_i, make the complete-data model essentially linear and Gaussian, which speeds up the M-step. Also note that we include the combined measurement error, U_i, instead of the "efficient" stock price, X_i. The combined error is a standard normal random variable, conditional on q_i, because $q_i = 0$ or 1. This representation will lead to a better performing EM-algorithm particularly when the magnitude of microstructure noises is small.[5]

The complete-data model's log-likelihood function allows the jump intensities – $\lambda_{x,op}$, $\lambda_{x,cl}$ and λ_y – to be separated from other parameters. This feature can be utilized to simplify the estimation problem. Denote all other parameters by χ and use the variables without the subscript i to represent the entire time series of those variables; for example, Y stands for $\{Y_0, ..., Y_n\}$. As shown in Appendix A, the complete-data log-likelihood function can be decomposed into three parts:

$$\mathcal{L}\left(Y, U, q, \Delta N | \, \chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}\right) = \mathcal{L}_1\left(Y | \, U, q, \Delta N, \chi\right) + \mathcal{L}_2(\Delta N | \lambda_{x,op}, \lambda_{x,cl}) + \mathcal{L}_3(q | \lambda_y) \tag{23}$$

[4] For a general introduction to the MCEM algorithm, see for instance Wei and Tanner (1990).

[5] The intuition is that the measurement-error based representation is less informative on the measurement-error parameters. It is well-known in the EM literature that the more informative the complete data is on the model parameters, the slower the EM-algorithm.

where

$$
\begin{aligned}
\mathcal{L}_1\left(Y|\,U, q, \Delta N, \chi\right) &= \sum_{i=1}^{n}\left(-\ln\sigma_i(\chi) - \frac{[Y_i - \mu_i(\chi)]^2}{2\sigma_i^2(\chi)}\right)\\
\mathcal{L}_2(\Delta N|\lambda_{x,op}, \lambda_{x,cl}) &= \sum_{t_i\in\mathcal{T}_{op}}\left[\Delta N_i \ln(\lambda_{x,op}\Delta t_i) - \lambda_{x,op}\Delta t_i\right] + \sum_{t_i\notin\mathcal{T}_{op}}\left[\Delta N_i \ln(\lambda_{x,cl}\Delta t_i) - \lambda_{x,cl}\Delta t_i\right]\\
\mathcal{L}_3(q|\lambda_y) &= \sum_{i=1}^{n}\left[q_i \ln\lambda_y + (1-q_i)\ln(1-\lambda_y)\right]\\
\mu_i(\chi) &= q_i\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_i + (1-q_i)\sigma_y U_i + \mu_x\Delta t_i + \mu_{J_x}\Delta N_i\\
&\quad +(1-\rho\Delta t_i)\left[Y_{i-1} - q_{i-1}\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_{i-1} - (1-q_{i-1})\sigma_y U_{i-1}\right]\\
\sigma_i^2(\chi) &= \sigma_{x,t_i}^2\Delta t_i + \Delta N_i\sigma_{J_x,t_i}^2.
\end{aligned}
$$

For the E-step of the EM algorithm, we need to evaluate the conditional expectation of the complete-data log-likelihood in (23). To do this we need to evaluate expectations of the following form:

$$E\left[f(Y_i, Y_{i-1}, q_i, q_{i-1}, \Delta N_i, U_i, U_{i-1}; \chi, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y)|\mathcal{D}_n, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right]$$

The localized particle filter described in section 2.2 can be used to compute this quantity. We run the filter using the parameters $(\chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y)$ to generate the particle set that represents the smoothed distribution for $(Y_i, Y_{i-1}, q_i, q_{i-1}, \Delta N_i, U_i, U_{i-1})$. The m-th particle is denoted by $\left(y_{i|n}^{(m)}, y_{i-1|n}^{(m)}, q_{i|n}^{(m)}, q_{i-1|n}^{(m)}, \Delta N_{i|n}^{(m)}, U_{i|n}^{(m)}, U_{i-1|n}^{(m)}\right)$. Thus, the expectation can be approximated by the sample average as follows:

$$
\begin{aligned}
&E\left[f(Y_i, Y_{i-1}, q_i, q_{i-1}, \Delta N_i, U_i, U_{i-1}; \chi, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y)|\mathcal{D}_n, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right]\\
\approx\;& \frac{1}{M}\sum_{m=1}^{M} f\left(y_{i|n}^{(m)}, y_{i-1|n}^{(m)}, q_{i|n}^{(m)}, q_{i-1|n}^{(m)}, \Delta N_{i|n}^{(m)}, U_{i|n}^{(m)}, U_{i-1|n}^{(m)}; \chi, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y\right). \qquad (24)
\end{aligned}
$$

When the sample size n is large, undesirable Monte-Carlo noise will be introduced by the use of the smoothed distribution. Intuitively, the particle filter always adapts to the newest observation, and thus its representation of the distant past is bound to be poor. Cappe and Moulines (2005) suggest to use the information only up to $i + L$ when computing any quantity that involves the unobserved state variable at time i. The rationale is the forgetting property expected of the dynamic system; that is, for large enough L, the distribution for the unobserved state variable at time i conditional on the information up to $i + L$ will be almost identical to that conditional on the entire sample.[6] Cappe and Moulines (2005) thus propose to use fixed-lag smoothing by using information only up to $i + L$. They present examples in which the bias induced by fixed-lag

[6] The practical filter in the MCMC algorithm of Polson, *et al* (2006) in effect uses the same rationale.

smoothing is minimal but the reduction in the Monte-Carlo error is dramatic. Adopting fixed-lag smoothing leads to our approximation as follows:

$$\begin{aligned}
& E\left[f(Y_i, Y_{i-1}, q_i, q_{i-1}, \Delta N_i, U_i, U_{i-1}; \chi, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y)|\mathcal{D}_n, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right] \\
\approx \; & E\left[f(Y_i, Y_{i-1}, q_i, q_{i-1}, \Delta N_i, U_i, U_{i-1}; \chi, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y)|\mathcal{D}_{(i+L)\wedge n}, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right] \\
\approx \; & \frac{1}{M}\sum_{m=1}^{M} f\left(y^{(m)}_{i|(i+L)\wedge n}, y^{(m)}_{i-1|(i+L)\wedge n}, q^{(m)}_{i|(i+L)\wedge n}, q^{(m)}_{i-1|(i+L)\wedge n}, \Delta N^{(m)}_{i|(i+L)\wedge n},\right. \\
& \left. U^{(m)}_{i|(i+L)\wedge n}, U^{(m)}_{i-1|(i+L)\wedge n}; \chi, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y\right) \qquad (25)
\end{aligned}$$

Applying this procedure to $\mathcal{L}_1\left(Y|\, U, q, \Delta N, \chi\right)$ yields

$$\begin{aligned}
& E\left[\mathcal{L}_1(Y|\, U, q, \Delta N, \chi)|\mathcal{D}_n, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right] \\
= \; & \sum_{i=1}^{n} E\left(-\ln\sigma_i(\chi) - \frac{[Y_i - \mu_i(\chi)]^2}{2\sigma_i(\chi)^2}\right|\left.\mathcal{D}_n, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right) \\
\approx \; & \sum_{i=1}^{n} E\left(-\ln\sigma_i(\chi) - \frac{[Y_i - \mu_i(\chi)]^2}{2\sigma_i(\chi)^2}\right|\left.\mathcal{D}_{(i+L)\wedge n}, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right) \\
\equiv \; & \hat{E}\left[\mathcal{L}_1\left(Y|\, U, q, \Delta N, \chi\right)|\mathcal{D}_n, \chi', \lambda'_{x,op}, \lambda'_{x,cl}, \lambda'_y\right] \qquad (26)
\end{aligned}$$

The expectation operator $\hat{E}(\cdot)$ denotes the expected value computed with the particle filter and using fixed-lag smoothing. One can similarly approximate the conditional expectations of $\mathcal{L}_2(\Delta N|\lambda_{x,op}, \lambda_{x,cl})$ and $\mathcal{L}_3(q|\lambda_y)$.

The MCEM algorithm can be summarized as follows: (1) Set some initial parameter values, $\left(\chi^{(0)}, \lambda^{(0)}_{x,op}, \lambda^{(0)}_{x,cl}, \lambda^{(0)}_y\right)$; (2) Repeat the following E- and M-steps until convergence.

- **E-step:** Run the localized particle filter at the parameter values $\left(\chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right)$. Compute (1) $\hat{E}\left[\mathcal{L}_1\left(Y|\, U, q, \Delta N, \chi\right)|\mathcal{D}_n, \chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right]$,
(2) $\hat{E}\left[\mathcal{L}_2\left(\Delta N|\lambda_{x,op}, \lambda_{x,cl}\right)|\, \mathcal{D}_n, \chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right]$, and
(3) $\hat{E}\left[\mathcal{L}_3\left(q|\lambda_y\right)|\, \mathcal{D}_n, \chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right]$.

- **M-step:** Maximize the conditional expected value of the complete-data log-likelihood function obtained in the E-step. The decomposition in (23) suggests that the M-step can be performed separately.

$$\begin{aligned}
\chi^{(k)} &= arg\max_{\chi} \hat{E}\left[\mathcal{L}_1\left(Y|\, U, q, \Delta N, \chi\right)|\mathcal{D}_n, \chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right] \\
(\lambda^{(k)}_{x,op}, \lambda^{(k)}_{x,cl}) &= arg\max_{(\lambda_{x,op}, \lambda_{x,cl})} \hat{E}\left[\mathcal{L}_2\left(\Delta N|\lambda_{x,op}, \lambda_{x,cl}\right)|\, \mathcal{D}_n, \chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right] \\
\lambda^{(k)}_y &= arg\max_{\lambda_y} \hat{E}\left[\mathcal{L}_3\left(q|\lambda_y\right)|\, \mathcal{D}_n, \chi^{(k-1)}, \lambda^{(k-1)}_{x,op}, \lambda^{(k-1)}_{x,cl}, \lambda^{(k-1)}_y\right]
\end{aligned}$$

In the above, $\lambda_{x,op}^{(k)}, \lambda_{x,cl}^{(k)}$ and $\lambda_y^{(k)}$ have the closed-form solutions expressed in terms of some conditional sufficient statistics. However, $\chi^{(k)}$ needs to be solved for numerically because the conditional expectation of $\mathcal{L}_1\left(Y \mid U, q, \Delta N, \chi\right)$ cannot be similarly simplified.

Since one item in the M-step (i.e., $\chi^{(k)}$) calls for repeated runs through a particle filter with the same model parameters, the above MCEM algorithm will be quite inefficient. A remedial device becomes available by considering the so-called generalized EM algorithm, meaning that one need not actually maximize the conditional expected value of the complete-data log-likelihood. As long as a parameter update improves the conditional expected value of the complete-data log-likelihood, the iterative system will still give rise to the desirable result.

We group the parameters in χ into three subsets – χ_μ consisting of the parameters affecting the conditional mean of Y_i, χ_{σ_x} consisting of the parameters determining the conditional variance process ($\sigma_{x,t}$), and $\chi_{\sigma_{J_x}}$ consisting of the jump volatility parameters (i.e., $\sigma_{J_x,op}$ and $\sigma_{J_x,cl}$). First, we optimize L_1 over χ_μ at some given values for χ_{σ_x} and $\chi_{\sigma_{J_x}}$. Optimizing over χ_{σ_x} or $\chi_{\sigma_{J_x}}$ by fixing the other parameter values turns out to still require running through the particle filter repeatedly. the reason is that χ_{σ_x} and $\chi_{\sigma_{J_x}}$ are not separable in L_1. We need to extend the complete-data space to achieve a separation and thus yield a more efficient scheme.

Decompose the innovation of X_i into the diffusion and jump components, denoted by $Z_i^C \equiv \sigma_{x,t_i}^2 \Delta W_i$ and $Z_i^J \equiv J_i \Delta N_i$, respectively. They are obviously independent of each other and with normal distributions: $Z_i^C \sim N(0, \sigma_{x,t_i}^2 \Delta t_i)$ and $Z_i^J \sim N(0, \Delta N_i \sigma_{J_x,t_i}^2)$. In Appendix B, we show that when we extend the complete data to include both Z_i^C and Z_i^J, the complete-data log-likelihood function, with respect to χ_{σ_x} and $\chi_{\sigma_{J_x}}$, can be decomposed nicely. It becomes a sum of two functions – $L_C(Y, U, q, \Delta N, Z^C, Z^J; \chi_{\sigma_x})$ and $L_J(Y, U, q, \Delta N, Z^C, Z^J; \chi_{\sigma_{J_x}})$. Moreover, with the extended complete data space, both χ_{σ_x} and $\chi_{\sigma_{J_x}}$ can be taken out of their respective conditional expectations, and therefore one only needs to run through the particle filter once. Optimization over χ_{σ_x} and $\chi_{\sigma_{J_x}}$ can thus be carried out efficiently.

Our more efficient MCEM algorithm can be summarized as follows: (1) Set some initial parameter values, $\left(\chi_\mu^{(0)}, \chi_{\sigma_x}^{(0)}, \chi_{\sigma_{J_x}}^{(0)}, \lambda_{x,op}^{(0)}, \lambda_{x,cl}^{(0)}, \lambda_y^{(0)}\right)$; (2) Repeat the following E- and M-steps until convergence.[7]

- **E1-step:** Run the particle filter using the parameter values $\left(\chi_\mu^{(k-1)}, \chi_{\sigma_x}^{(k-1)}, \chi_{\sigma_{J_x}}^{(k-1)}, \lambda_{x,op}^{(k-1)}, \lambda_{x,cl}^{(k-1)}, \lambda_y^{(k-1)}\right)$. Store all the conditional sufficient statistics needed for the M1-step.
- **M1-step:** Maximize the expected value of the complete-data log-likelihood function over the subset of parameters, $(\chi_\mu, \lambda_{x,op}, \lambda_{x,cl}, \lambda_y)$. This maximization yields closed-form solutions

[7] Our algorithm is in effect a simulated version of the space alternating generalized EM algorithm of Fessler and Hero (1994).

expressed in terms of the conditional sufficient statistics computed in the E1-step.

$$\begin{aligned}
&\chi_\mu^{(k)} = arg\max_{\chi_\mu} \\
&\hat{E}\left[\mathcal{L}_1\left(Y|U,q,\Delta N,\chi_\mu,\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right)\middle|\,\mathcal{D}_n,\chi_\mu^{(k-1)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)},\lambda_{x,op}^{(k-1)},\lambda_{x,cl}^{(k-1)},\lambda_y^{(k-1)}\right], \\
&(\lambda_{x,op}^{(k)},\lambda_{x,cl}^{(k)}) = arg\max_{(\lambda_{x,op},\lambda_{x,cl})} \\
&\hat{E}\left[\mathcal{L}_2\left(\Delta N|\lambda_{x,op},\lambda_{x,cl}\right)\middle|\,\mathcal{D}_n,\chi_\mu^{(k-1)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)},\lambda_{x,op}^{(k-1)},\lambda_{x,cl}^{(k-1)},\lambda_y^{(k-1)}\right], \\
&\lambda_y^{(k)} = arg\max_{\lambda_y}\hat{E}\left[\mathcal{L}_3\left(q|\lambda_y\right)\middle|\,\mathcal{D}_n,\chi_\mu^{(k-1)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)},\lambda_{x,op}^{(k-1)},\lambda_{x,cl}^{(k-1)},\lambda_y^{(k-1)}\right].
\end{aligned}$$

- **E2-step:** Run the particle filter using the parameter values, $(\chi_\mu^{(k)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)},\lambda_{x,op}^{(k)},\lambda_{x,cl}^{(k)},\lambda_y^{(k)})$. Store all the conditional sufficient statistics needed for the M2-step.

- **M2-step:** Optimize the expected value of the complete-data log-likelihood function defined specifically for χ_{σ_x} and $\chi_{\sigma_{J_x}}$ where the complete data space has been extended to (Z^C,Z^J,Y,U,q,dN).

$$\begin{aligned}
\chi_{\sigma_x}^{(k)} &= arg\max_{\chi_{\sigma_x}}\hat{E}\left[L_C(Y,U,q,\Delta N,Z^C,Z^J;\chi_{\sigma_x})\mid\mathcal{D}_n,\chi_\mu^{(k)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)},\lambda_{x,op}^{(k)},\lambda_{x,cl}^{(k)},\lambda_y^{(k)}\right], \\
\chi_{\sigma_{J_x}}^{(k)} &= arg\max_{\chi_{\sigma_{J_x}}}\hat{E}\left[L_J(Y,U,q,\Delta N,Z^C,Z^J;\chi_{\sigma_{J_x}})\mid\mathcal{D}_n,\chi_\mu^{(k)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)},\lambda_{x,op}^{(k)},\lambda_{x,cl}^{(k)},\lambda_y^{(k)}\right].
\end{aligned}$$

In the M2-step, $\chi_{\sigma_{J_x}}^{(k)}$ has a closed-form solution. If the conditional variance process $(\sigma_{x,t})$ were non-stochastic, $\chi_{\sigma_x}^{(k)}$ would also have a closed-form. But in our empirical analysis presented later, we have adopted a GARCH-type conditional variance process, which means that getting $\chi_{\sigma_x}^{(k)}$ requires iterations. Such iterations, however, do not need repeated runs through the particle filter. In short, this optimization is equivalent to estimating a GARCH-type time-series model with n observations. Better still, one only need to, by the generalized EM algorithm, take a couple of iterations for this part of the M2-step.

In order to conduct statistical inference, we need to get an estimate for the asymptotic covariance matrix. The usual approach to computing the outer product of the observed-data individual scores is not applicable here because of the incomplete-data structure. However, we describe in Appendix C that there is an asymptotically equivalent covariance estimator that only uses the smoothed complete-data individual scores, which in turn can easily be approximated with our particle filter using fixed-lag smoothing.

3 Disentangling genuine jumps from microstructure noise

3.1 Specification of the volatility model

In order to implement the estimation method on real data, we need to be specific about the volatility dynamic. The empirical success of the GARCH model in handling daily return volatility motivates our model choice. We assume a GARCH-like volatility dynamic for the daily variance, but the variance innovation comes from the unanticipated change in the previous day's realized variance rather than from the daily return innovation. Specifically, we deal with $d = 1, ..., D$ days in the sample and denote the observations on day d with the set $\mathcal{I}_d$. Each day is assumed to last from the close of the previous day's trading session until close of the current day's trading session. For each day, the annualized realized variance RV_d is defined as

$$RV_d = \frac{1}{\sum 1_{\{t_i \in \mathcal{I}_d \bigcap \mathcal{T}_{op}\}}} \sum_{t_i \in \mathcal{I}_d \bigcap \mathcal{T}_{op}} \frac{(\Delta \ln s_i)^2}{\Delta t_i}. \tag{27}$$

The daily variance is assumed to evolve according to

$$h_d = \alpha_0 + \alpha_1 h_{d-1} + \beta_1 (RV_{d-1} - E_{d-2}(RV_{d-1})), \tag{28}$$

where

$$E_{d-1}(RV_d) \approx h_d + \lambda_{x,op} \sigma^2_{J_x,op} + \frac{2}{\Delta t}(\sigma^2_y + \lambda_y \sigma^2_{J_y}),$$

a result from ignoring the price discretization error and all terms at or above the order of $(\Delta t)^2$.

Furthermore, we assume that the local variance on day d remains at h_d throughout the entire trading session. However, we account for the well-known fact that information arrives at a different rate when markets are closed vis-a-vis open. The closed session variance differs from the open session by a constant ratio, φ; that is, for $t_i \in \mathcal{I}_d$ and $d = 1, ..., D$,

$$\sigma^2_{x,t_i} = h_d 1_{t_i \in \mathcal{T}_{op}} + \varphi h_d 1_{t_i \notin \mathcal{T}_{op}} \tag{29}$$

To summarize, the parameter vector defining $\sigma_{x,t}$ is $\chi_{\sigma_x} = (\alpha_0, \alpha_1, \beta_1, h_1, \varphi)$, where the initial value of the daily variance process, h_1, has been treated as an unknown parameter.

3.2 Empirical Results

Frequent small jumps in the stock price may give rise to an appearance of infrequent large jumps if one uses lower-frequency data to conduct the analysis. The reason is fairly clear. Both frequent small and infrequent large jumps can force the daily stock return to be skewed and heavy-tailed. For example, a stock that is subject to an average of 25 relative large jumps a year will, loosely speaking, have 10% daily returns coming from the heavy-tailed distribution. In contrast, a stock that is subject to many small jumps within a day will after aggregation also have some of its daily returns to exhibit heavy tails. The only way to detect the true nature of jumps with a reasonable level of confidence is to utilize high-frequency data. In a way, frequent small jumps can reveal themselves to be distinctively different from infrequent large jumps.[8] Being able to extract the "efficient" stock value from the observed stock price contaminated by microstructure noises is critical. Otherwise, one can easily mistake microstructure noises as frequent small jumps.

We use intra-day data from the NYSE TAQ database to extract a price series at the sampling frequency of 5 minutes. Specifically, for each point on a 5-minute time grid between 9.30 and 16.00, we store the transaction prices closest to the time grid.

We use 100 particles (i.e., $M = 100$) for our particle filter when microstructure noises are allowed. For the fixed-lag smoothing algorithm, we set $L = 5$. For the asymptotic variance calculations we use 5000 particles (i.e., $M = 5000$).

Tables 1-4 report the results for the IBM data in 2004 with several sampling frequencies – 5 minutes, 10 minutes, 30 minutes and 1 hour. For each sampling frequency, we estimate the model with and without jumps in the "efficient" stock price. We also investigate the consequences of different assumptions on microstructure noises. In all cases, we have set the mean-reversion parameter to zero (i.e., $\rho = 0$).

There are several points to be noted in these tables. First, the intuition mentioned earlier seems to bear out in these results; that is, sampling at a lower frequency leads to estimates that indicate a lower jump intensity but a larger jump magnitude. Let us turn attention to M6, which is a model allowing for asset jumps and heavy-tailed microstructure noises. As we move the sampling frequency from hourly to every 10 minutes, the jump intensity estimate for the trading session (i.e, $\lambda_{x,op}$) goes up from around 5600 to 14100, indicating that the average number of jumps during one trading session (9:30am to 4:00pm Eastern Standard Time) increases from roughly 5.4 to 13.6. At the same time, we find the estimate for the jump size volatility for the trading session (i.e., $\sigma_{J_x,op}$) decreases from 0.003 to 0.0017. In conclusion, if one wants to detect small frequent jumps, it is important to sample at a higher frequency.

Our results also underline another point raised earlier; that is, the importance of allowing for microstructure noises when high-frequency data are used to estimate jumps. Tables 2 and 3 show that at the hourly and 30-minute sampling frequencies, microstructure noises have minor effects on the estimates. However, Table 1 indicates that microstructure noises are indeed important at the 10-minute sampling frequency. When one ignores microstructure noises (i.e., M4), the estimate for the jump intensity is $\lambda_{x,op} = 22900$. When they are allowed but are forced to be normally distributed (i.e., M5), the jump intensity estimate goes down to $\lambda_{x,op} = 20900$. An even larger decrease can be seen when microstructure noises are allowed to be heavy-tailed (i.e., M6). In that case, $\lambda_{x,op} = 14100$. Thus, ignoring microstructure noises in estimation will give rise to an

[8] When the true data generating process is of frequent small jumps, using lower-frequency data to estimate the jump intensity and magnitude will lead to large estimator uncertainty and thus may give rise to an appearance of a low jump intensity coupled with a large jump magnitude. We are definitely not claiming that infrequent large jumps and frequent small jumps will always leave the same signature when one uses, say, daily data.

appearance of more jumps, leading to an overestimation of the jump intensity by roughly 50%.

Our results are by and large in line with the literature in other aspects. First, there are marked differences between the trading and closed sessions. The diffusion part of the asset movement seems to be less active during the closed session with the ratio of activities, φ, estimated to be around 40% to 50%. Jumps in the closed session seem to be much less frequent but larger in magnitude. One possible explanation is that important corporate news are announced when the market is closed. Thus, the estimates simply reflect the industry practice in handling news release. Second, we find strong evidence for volatility clustering, with the parameter controlling the volatility persistence, α_1, to be around 0.8-0.9.

To make sure that the results are not unique to year 2004, we repeat the estimation for 2002, 2003 and 2005. To conserve space, Table 5 only reports the estimates for the jump intensity in the open period, i.e., $\lambda_{X,op}$ and the corresponding jump size's standard deviation, i.e., $\sigma_{JX,op}$. The numbers are consistent with the findings for 2004. First, accounting for microstructure noises is important at high frequency and becomes immaterial as the sampling frequency decreases. Second, as the sampling frequency increases, jumps appear to be more frequent but smaller in magnitude.

The filtering technique allows one to go beyond parameter estimation; for example, one can ask how the observed stock price differs from the "efficient" price. At the ML parameter estimates, one can run through the particle filter to come up the series of either filtered or smoothed "efficient" stock prices.[9] We pick two days (January 2, 2004 and May 25, 2004) to show the magnitude of microstructure noises. We use the full model (i.e., M6) on the IBM prices sampled every 5 minutes. Figures 1 and 2 show that the observed stock prices stay within 5 cents of the filtered prices, suggesting that the microstructure effect is no more than 5 cents in these two days. The smoothed prices, however, tell a different story. The results indicate that the difference between the observed and smoothed prices was much larger and in fact was sometimes larger than 10 cents in these two days. Large differences also appear to cluster towards the end of a trading session.

[9] The filtered price means conditioning on the observed prices only up to the time point of interest, whereas the smoothed price refers to conditioning on the entire sample of the observed prices.

Appendix A: The complete-data log-likelihood function

First note the following facts about the complete-data representation:

C1: q_i are *i.i.d.* Bernoulli draws with the parameter $P(q_i = 1) = \lambda_y$. Thus, the likelihood of q_i only depends on λ_y.

C2: ΔN_i are independent Poisson draws with parameters $\lambda_{x,t_i}\Delta t_i$. Thus, the likelihood of ΔN_i only depends on λ_{x,t_i}.

C3: Conditional on q_i, the distribution of U_i does not depend on the model parameters because it is a standard normal random variable.

C4: Using the definition of U_i, equation (5) can be rewritten as

$$Y_i = X_i + q_i\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_i + (1 - q_i)\sigma_y U_i \tag{30}$$

Express X_i in terms of other variables using (30) and then substitute it into (6). Conditioning on $(U_i, q_i, \Delta N_i)$, we obtain the following transition equation for Y_i:

$$\begin{aligned} Y_i &= q_i\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_i + (1 - q_i)\sigma_y U_i \\ &\quad +(1 - \rho\Delta t_i)\left[Y_{i-1} - q_{i-1}\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_{i-1} - (1 - q_{i-1})\sigma_y U_{i-1}\right] \\ &\quad +\mu_x\Delta t_i + \sigma_x\Delta W_i + J_i\Delta N_i \\ &= q_i\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_i + (1 - q_i)\sigma_y U_i \\ &\quad +(1 - \rho\Delta t_i)\left[Y_{i-1} - q_{i-1}\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_{i-1} - (1 - q_{i-1})\sigma_y U_{i-1}\right] \\ &\quad +\mu_x\Delta t_i + \mu_{J_x}\Delta N_i + \sqrt{\sigma_x^2\Delta t_i + \Delta N_i\sigma_{J_x,t_i}^2} \times \upsilon_i \end{aligned} \tag{31}$$

where υ_i is a standard normal random variable independent of $(U_i, q_i, \Delta N_i)$. Equation (31) shows that conditional on $(U_i, q_i, \Delta N_i)$, Y_i follows an autoregressive process with a time-varying mean and variance. Moreover, the innovation of the autoregressive system is normally distributed. Needless to say, the likelihood of Y_i, conditional on $(U_i, q_i, \Delta N_i)$, does not depend on λ_y, $\lambda_{x,op}$ or $\lambda_{x,cl}$.

C1-C4 allow us to decompose the complete-data log-likelihood function into three parts with each governed by different disjoint subsets of parameters.

$$\begin{aligned} & f\left(Y, U, q, \Delta N|\, \chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}\right) \\ = \;& f\left(Y|\, U, q, \Delta N, \chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}\right) f\left(U|\, q, \Delta N, \chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}\right) \\ & \times f(\Delta N|\chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}) f(q|\chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}) \\ = \;& f\left(Y|\, U, q, \Delta N, \chi\right) f(\Delta N|\lambda_{x,op}, \lambda_{x,cl}) f(q|\lambda_y) f(U|q) \\ \propto \;& f\left(Y|\, U, q, \Delta N, \chi\right) f(\Delta N|\lambda_{x,op}, \lambda_{x,cl}) f(q|\lambda_y). \end{aligned} \tag{32}$$

The above derivation utilizes the fact that conditional on q_i, U_i is always a standard normal random variable, and thus the likelihood function does not depend on any model parameter and amounts to an irrelevant constant. The above result in turn implies

$$\mathcal{L}\left(Y, U, q\Delta N \middle| \, \chi, \lambda_y, \lambda_{x,op}, \lambda_{x,cl}\right) = \mathcal{L}_1\left(Y \middle| \, U, q, \Delta N, \chi\right) + \mathcal{L}_2(\Delta N | \lambda_{x,op}, \lambda_{x,cl}) + \mathcal{L}_3(q|\lambda_y)$$

where

$$\begin{aligned}
\mathcal{L}_1\left(Y \middle| \, U, q, \Delta N, \chi\right) &= \sum_{i=1}^{n}\left(-\ln \sigma_i(\chi) - \frac{[Y_i - \mu_i(\chi)]^2}{2\sigma_i^2(\chi)}\right) \\
\mathcal{L}_2(\Delta N|\lambda_{x,op}, \lambda_{x,cl}) &= \sum_{t_i \in \mathcal{T}_{op}} [\Delta N_i \ln(\lambda_{x,op}\Delta t_i) - \lambda_{x,op}\Delta t_i] + \sum_{t_i \notin \mathcal{T}_{op}} [\Delta N_i \ln(\lambda_{x,cl}\Delta t_i) - \lambda_{x,cl}\Delta t_i] \\
\mathcal{L}_3(q|\lambda_y) &= \sum_{i=1}^{n} [q_i \ln \lambda_y + (1 - q_i)\ln(1 - \lambda_y)] \\
\mu_i(\chi) &= q_i\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_i + (1 - q_i)\sigma_y U_i + \mu_x \Delta t_i + \mu_{J_x}\Delta N_i \\
&\quad + (1 - \rho\Delta t_i)\left[Y_{i-1} - q_{i-1}\sqrt{\sigma_y^2 + \sigma_{J_y}^2}U_{i-1} - (1 - q_{i-1})\sigma_y U_{i-1}\right] \\
\sigma_i^2(\chi) &= \sigma_{x,t_i}^2 \Delta t_i + \Delta N_i \sigma_{J_x,t_i}^2.
\end{aligned}$$

Appendix B: Separating diffusion innovations from jumps

Extend the complete-data space to $(Z^C, Z^J, Y, U, q, \Delta N)$. The two new elements, Z_i^C and Z_i^J, decompose the innovation in the "efficient" stock price, X_i, into the diffusion and jump components. Define $Z^C = (Z_i^C; i = 1, ..., n)$ and $Z^J = (Z_i^J; i = 1, ..., n)$. Clearly, Z_i^C and Z_i^J are independent over time and with each other. They are also normally distributed as follows:

$$Z_i^C \sim N(0, \sigma_{x,t_i}^2 \Delta t_i) \text{ and } Z_i^J \sim N(0, \Delta N_i \sigma_{J_x,t_i}^2)$$

Denote by $\mathcal{F}_{i-1}$ the information generated by the complete data up to $i-1$. The extended complete-data likelihood function (focusing on the parameters of interest, χ_{σ_x} and $\chi_{\sigma_{J_x}}$) can be simplified to

$$\begin{aligned}
f(Y, U, q, \Delta N, Z^C, Z^J \mid \chi_{\sigma_x}, \chi_{\sigma_{J_x}}) &\propto \prod_{i=1}^{n} f(Z_i^C, Z_i^J \mid \mathcal{F}_{i-1}, \chi_{\sigma_x}, \chi_{\sigma_{J_x}}) \\
&= \prod_{i=1}^{n} f(Z_i^C \mid \mathcal{F}_{i-1}, \chi_{\sigma_x}) f(Z_i^J \mid \mathcal{F}_{i-1}, \chi_{\sigma_{J_x}}).
\end{aligned}$$

The first relationship utilizes the fact that the terms unrelated to the two parameters of interest can be dropped. The second equality of course follows from the conditional independence of Z_i^C

and Z_i^J. Since the extended complete-data log-likelihood function is separable in terms of χ_{σ_x} and $\chi_{\sigma_{J_x}}$, it can be written as a sum of $L_C(Y,U,q,\Delta N,Z^C,Z^J;\chi_{\sigma_x})$ and $L_J(Y,U,q,\Delta N,Z^C,Z^J;\chi_{\sigma_{J_x}})$, after dropping the irrelevant constant. Hence

$$L_C(Y,U,q,\Delta N,Z^C,Z^J;\chi_{\sigma_x}) = \sum_{i=1}^{n}\left(-\ln\sigma_{x,t_i}(\chi_{\sigma_x}) - \frac{(Z_i^C)^2}{2\sigma^2_{x,t_i}(\chi_{\sigma_x})\Delta t_i}\right)$$

Although σ_{x,t_i} is allowed to be stochastic, our maintained assumption requires it to be measurable with respect to $\mathcal{D}_{i-1}$. Taking the conditional expectation with respect to the smoothed distribution for the complete data (suppressing some conditioning parameters for notational simplicity) gives rise to

$$\begin{aligned} & E\left[L_C(Y,U,q,\Delta N,Z^C,Z^J;\chi_{\sigma_x})\middle|\,\mathcal{D}_n,\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right] \\ = & \sum_{i=1}^{n}\left(-\ln\sigma_{x,t_i}(\chi_{\sigma_x}) - \frac{E\left[(Z_i^C)^2\middle|\,\mathcal{D}_n,\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right]}{2\sigma^2_{x,t_i}(\chi_{\sigma_x})\Delta t_i}\right)\end{aligned}$$

Moreover,

$$\begin{aligned} & E\left[(Z_i^C)^2\middle|\,\mathcal{D}_n,\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right] \\ = & E\left\{E\left[(Z_i^C)^2 \mid Y,U,q,\Delta N,\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right]\middle|\,\mathcal{D}_n,\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right\} \\ \approx & \frac{1}{M}\sum_{m=1}^{M} E\left[(Z_i^C)^2\middle|\,Y_i^{(m)},\mu_i^{(m)},\Delta N_i^{(m)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right]\end{aligned}$$

The above expression turns out to have a closed-form solution. First note that

$$\left(\begin{array}{c} Y_i-\mu_i \\ Z_i^C \end{array}\right) = \left(\begin{array}{c} Z_i^C+Z_i^J \\ Z_i^C \end{array}\right)\Bigg|\Delta N_i \sim N\left(\left[\begin{array}{c}0\\0\end{array}\right],\left[\begin{array}{cc}\sigma^2_{x,t_i}\Delta t_i+\Delta N_i\sigma^2_{J_x,t_i} & \sigma^2_{x,t_i}\Delta t_i \\ \sigma^2_{x,t_i}\Delta t_i & \sigma^2_{x,t_i}\Delta t_i\end{array}\right]\right)$$

which implies

$$\begin{aligned} Z_i^C\big|\,(Y_i,\mu_i,\Delta N_i) & \sim N(\mu^{(k-1)},\eta^{(k-1)}) \\ \mu^{(k-1)} & = \frac{(\sigma_{x,t_i}^{(k-1)})^2\Delta t_i}{(\sigma_{x,t_i}^{(k-1)})^2\Delta t_i+\Delta N_i(\sigma_{J_x,t_i}^{(k-1)})^2}(Y_i-\mu_i) \\ (\eta^{(k-1)})^2 & = (\sigma_{x,t_i}^{(k-1)})^2\Delta t_i\left(1-\frac{(\sigma_{x,t_i}^{(k-1)})^2\Delta t_i}{(\sigma_{x,t_i}^{(k-1)})^2\Delta t_i+\Delta N_i(\sigma_{J_x,t_i}^{(k-1)})^2}\right).\end{aligned}$$

Thus,

$$E\left[(Z_i^C)^2\middle|\,Y_i^{(m)},\mu_i^{(m)},\Delta N_i^{(m)},\chi_{\sigma_x}^{(k-1)},\chi_{\sigma_{J_x}}^{(k-1)}\right] = [\mu^{(k-1)}]^2_{(i,m)}+[\eta^{(k-1)}]^2_{(i,m)}$$

where dependence on time and particle is reflected in the subscript (i, m). To summarize, we have the following approximation to one of the two components for the expected value of the extended complete-data log-likelihood:

$$\begin{aligned} & E\left(L_C(Y, U, q, \Delta N, Z^C, Z^J; \chi_{\sigma_x}) \middle| \mathcal{D}_n, \chi_{\sigma_x}^{(k-1)}, \chi_{\sigma_{J_x}}^{(k-1)} \right) \\ \approx\ & \sum_{i=1}^{n} \left(-\ln \sigma_{x,t_i}(\chi_{\sigma_x}) - \frac{\frac{1}{M}\sum_{m=1}^{M}\left([\mu^{(k-1)}]^2_{(i,m)} + [\eta^{(k-1)}]^2_{(i,m)} \right)}{2\sigma^2_{x,t_i}(\chi_{\sigma_x})\Delta t_i} \right) \end{aligned}$$

In a similar fashion, one can compute the other component of the expected value of the extended complete-data log-likelihood; that is, $E\left[L_J(Y, U, q, \Delta N, Z^C, Z^J; \chi_{\sigma_{J_x}}) \middle| \mathcal{D}_n, \chi_{\sigma_x}^{(k-1)}, \chi_{\sigma_{J_x}}^{(k-1)} \right]$.

Appendix C: Computing asymptotic standard errors

The usual way to compute the asymptotic standard error for the maximum likelihood estimate is to use the negative Hessian matrix or the inner product of the individual scores. But in our case, either one is not directly computable because the individual log-likelihood function, $\ln f(s_i; i = 1, \cdots, n \mid \theta)$ is highly irregular with respect to θ due to using the particle filter. An alternative estimator proposed by Duan and Fulop (2006b) can be applied to our setting, however, which uses the smoothed individual scores to compute the asymptotic error.

Denote by α_i the complete-data vector at i. In our case, $\alpha_i = (Y_i, q_i, \Delta N_i, U_i)$ as defined in section 2.3. The complete-data log-likelihood function, $\ln g(\alpha_i; i = 1, \cdots, n \mid \theta)$, can be expressed as

$$\ln g(\alpha_i; i = 1, \cdots, n \mid \theta) = \sum_{i=1}^{n} \ln g_i(\alpha_i \mid \alpha_{i-1}, \theta)$$

where $\ln g_i(\alpha_i \mid \alpha_{i-1}, \theta)$ is the complete-data individual log-likelihood function. Dempster, *et al* (1977) and Louis (1982) show that the observed-data score can be decomposed into the sum of smoothed individual scores:

$$S_n(\theta) = E\left(\frac{\partial \ln g(\alpha_i; i = 1, \cdots, n \mid \theta)}{\partial \theta} \middle| \mathcal{D}_n, \theta \right) = \sum_{i=1}^{n} a_i(\theta),$$

where $a_i(\theta) = E\left(\frac{\partial \ln g_i(\alpha_i|\alpha_{i-1},\theta)}{\partial\theta} \middle| \mathcal{D}_n, \theta \right)$. Note that the smoothed individual scores, $a_i(\theta)$'s, can be computed in a straightforward fashion within our particle filter using fixed-lag smoothing. Duan and Fulop (2006b) devise an estimator with the insight that the variance of the observed-data score equals the negative Hessian matrix when both are evaluated at the true parameter value, θ_0. Then, they seek for an alternative way to approximate $Var(S_n(\theta_0))$. Their solution is to recognize that the smoothed individual scores are not martingale differences but the variance can be approximated with the Newey-West (1987) estimator. Assume that beyond some lags, say l, dependence among $a_i's$ becomes negligible. The alternative estimator for the asymptotic error is:

$$Var(S_n(\theta_0)) \approx \Omega_0 + \sum_{j=1}^{l} w(l)(\Omega_j + \Omega_j')$$

where

$$\begin{aligned} \Omega_j &= \sum_{i=1}^{n-j} a_i(\hat{\theta}) a_{i+j}(\hat{\theta})' \text{ and} \\ w(j) &= 1 - \frac{j}{l}. \end{aligned}$$

References

[1] Ait-Sahalia, Y., P. Mykland and L. Zhang, 2005a, A Tale of Two Time Scales: Determining Integrated Volatility with Noisy High-Frequency Data, *Journal of the American Statistical Association* 100, 1394–1411.

[2] Ait-Sahalia, Y., P. Mykland and L. Zhang, 2005b, How often to sample a continuous-time process in the presence of market microstructure noise, *Review of Financial Studies* 18, 351–416.

[3] Bandi, F. and J. Russell, 2006, Separating microstructure noise from volatility, *Journal of Financial Economics* 79, 655–692.

[4] Bates, D., 2000, Post-'87 Crash fears in S&P 500 futures options, *Journal of Econometrics* 94, 181–238.

[5] Cappe, O. and E. Moulines, 2005, On the Use of Particle Filtering for Maximum Likelihood Parameter Estimation, European Signal Processing Conference (EUSIPCO), Antalya, Turkey.

[6] Dempster, A.P., N.M. Laird and D.B. Rubin, 1977, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the American Statistical Association, Series B* 39, 1–38.

[7] Duan, J.C. and A. Fulop, 2006a, Estimating the Structural Credit Risk Model When Equity Prices Are Contaminated by Trading Noises, University of Toronto working paper.

[8] Duan, J.C. and A. Fulop, 2006b, A Stable Estimator for the Information Matrix under EM, University of Toronto working paper.

[9] Eraker, B., M. Johannes and N. Polson, 2003, The impact of jumps in volatility and returns, *Journal of Finance* 58, 1269–1300.

[10] Fessler, J.A. and A.O. Hero, 1994, Space alternating generalized expectation maximization algorithm, *IEEE Transactions in Image Processing* 42, 2664–2677.

[11] Louis, T.A., 1982, Finding the observed information matrix when using the EM algorithm, *Journal of the Royal Statistical Society, Series B* 44, 226–233.

[12] Newey, W.K. and K.D. West, 1987, A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix, *Econometrica* 55, 703–708.

[13] Pan, J., 2002, The jump-risk premia implicit in options: Evidence from an integrated time-series study, *Journal of Financial Economics* 63, 3–50.

[14] Pitt, M., 2002, Smooth particle filters likelihood evaluation and maximisation', University of Warwick working paper.

[15] Pitt, M. and N. Shephard, 1999, Filtering via simulation: auxiliary particle filter, *Journal of the American Statistical Association* 94, 590–599.

[16] Polson, N.G., J.R. Stroud and P. Muller, 2006, Practical Filtering with Sequential Parameter Learning, Working Paper, The Wharton School, University of Pennsylvania.

[17] Tauchen, G. and H. Zhou, 2005, Identifying Realized Jumps on Financial Markets, Working Paper, Duke University.

[18] Wei, G.C.G. and M.A. Tanner, 1990, A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms, *Journal of the American Statistical Association* 85, 699–704.

Table 1: 2004 IBM stock prices, 5-minute sampling frequency

	M1	M2	M3	M4	M5	M6
	Asset Diffusion Parameters					
μ_X	-0.119	-0.104	-0.0341	-0.276	-0.265	-0.246
	(-0.5714)	(-0.5059)	(-0.1707)	(-2.399)	(-2.427)	(-2.237)
α_0	0.00912	0.00967	0.0149	0.0114	0.00935	0.0108
	(10.57)	(11.92)	(11.67)	(9.573)	(9.601)	(9.186)
α_1	0.92	0.903	0.841	0.771	0.746	0.749
	(117.1)	(105.1)	(59.24)	(33.66)	(30.51)	(28.74)
β_1	0.277	0.272	0.211	0.179	0.182	0.16
	(26.23)	(25.88)	(18.23)	(14.13)	(14.38)	(13.04)
h_1	0.0972	0.0817	0.0743	0.0667	0.0543	0.0593
	(13.08)	(10.27)	(7.371)	(5.92)	(4.651)	(5.299)
φ	0.377	0.401	0.431	0.396	0.466	0.408
	(55.86)	(56.41)	(59)	(20.18)	(19.36)	(18.75)
	Asset Jump Parameters					
$\lambda_{X,op}$	0	0	0	3.7e+004	3.64e+004	2.37e+004
				(15.52)	(15.63)	(11.5)
$\lambda_{X,cl}$	0	0	0	11.3	9.95	12.7
				(1.745)	(1.618)	(1.802)
$\sigma_{JX,op}$	0	0	0	0.00127	0.00127	0.00132
				(40.73)	(41.22)	(30.87)
$\sigma_{JX,cl}$	0	0	0	0.0237	0.0247	0.0224
				(2.973)	(2.604)	(3.297)
μ_{JX}	0	0	0	4.51e-005	3.88e-005	5.73e-005
				(2.276)	(2.053)	(2.33)
	Measurement Error Parameters					
σ_Y	0	0.000262	1.22e-005	0	0.000254	0.000108
		(24.16)	(0.1108)		(24.32)	(3.941)
λ_Y	0	0	0.0594	0	0	0.1
			(16.39)			(10.4)
σ_{JY}	0	0	0.00179	0	0	0.0011
			(41.95)			(26.61)
Loglikelihood	80184.0	80235.5	80968.0	81273.7	81339.9	81488.0

The values inside parentheses are *t*-statistics.

M1 is the model without value jumps and without measurement errors.

M2 is the model without value jumps and with normally distributed measurement errors.

M3 is the model without value jumps and with heavy-tailed measurement errors.

M4-M6 correspond to M1-M3 except for allowing value jumps.

Table 2: 2004 IBM stock prices, 10-minute sampling frequency

	M1	M2	M3	M4	M5	M6
	Asset Diffusion Parameters					
μ_X	-0.118	-0.112	0.124	-0.265	-0.267	-0.236
	(-0.5679)	(-0.5411)	(0.6727)	(-2.353)	(-2.48)	(-2.142)
α_0	0.0132	0.0132	0.0147	0.0067	0.00578	0.00667
	(8.533)	(9)	(7.783)	(6.538)	(6.555)	(6.372)
α_1	0.874	0.862	0.835	0.845	0.837	0.831
	(56.98)	(53.72)	(36.71)	(36.9)	(36.06)	(33.06)
β_1	0.249	0.245	0.175	0.162	0.162	0.147
	(16.93)	(16.59)	(11.07)	(11.47)	(11.55)	(10.68)
h_1	0.0976	0.0874	0.0756	0.0623	0.0581	0.0538
	(7.992)	(6.877)	(5.863)	(4.392)	(3.957)	(3.949)
φ	0.396	0.413	0.435	0.424	0.472	0.428
	(56.82)	(51.24)	(50.22)	(19.37)	(17.75)	(17.4)
	Asset Jump Parameters					
$\lambda_{X,op}$	0	0	0	2.37e+004	2.02e+004	1.34e+004
				(10.88)	(10.83)	(8.552)
$\lambda_{X,cl}$	0	0	0	12.2	10.7	14.4
				(1.876)	(1.786)	(1.863)
$\sigma_{JX,op}$	0	0	0	0.00157	0.00165	0.00177
				(28.42)	(28.13)	(22.35)
$\sigma_{JX,cl}$	0	0	0	0.0229	0.0236	0.0214
				(3.301)	(2.971)	(3.683)
μ_{JX}	0	0	0	5.6e-005	6.47e-005	8.52e-005
				(1.879)	(1.992)	(1.964)
	Measurement Error Parameters					
σ_Y	0	0.000298	9.3e-005	0	0.000322	0.000116
		(10.14)	(1.043)		(15.23)	(1.684)
λ_Y	0	0	0.0556	0	0	0.127
			(10.37)			(6.714)
σ_{JY}	0	0	0.00252	0	0	0.00126
			(28.63)			(16.46)
Loglikelihood	37352.1	37363.5	37673.6	37897.2	37921.1	37985.7

The values inside parentheses are t-statistics.

M1 is the model without value jumps and without measurement errors.

M2 is the model without value jumps and with normally distributed measurement errors.

M3 is the model without value jumps and with heavy-tailed measurement errors.

M4-M6 correspond to M1-M3 except for allowing value jumps.

Table 3: 2004 IBM stock prices, 30-minute sampling frequency

	M1	M2	M3	M4	M5	M6
	Asset Diffusion Parameters					
μ_X	-0.102	-0.102	-0.104	-0.27	-0.271	-0.25
	(-0.4751)		(-0.5594)	(-2.389)	(-2.411)	(-2.37)
α_0	0.0128	0.0128	0.0128	0.00534	0.00474	0.00477
	(4.49)		(4.197)	(3.278)	(3.172)	(3.058)
α_1	0.866	0.866	0.856	0.836	0.837	0.838
	(28.93)		(23.76)	(17.13)	(17.8)	(17.02)
β_1	0.137	0.137	0.113	0.102	0.0994	0.0943
	(6.986)		(5.704)	(5.438)	(5.419)	(5.19)
h_1	0.111	0.111	0.104	0.0563	0.0561	0.0529
	(4.193)		(3.63)	(2.355)	(2.31)	(2.27)
φ	0.418	0.418	0.417	0.478	0.509	0.479
	(48.66)		(34.09)	(13.92)	(11.77)	(11.18)
	Asset Jump Parameters					
$\lambda_{X,op}$	0	0	0	1.11e+004	1.12e+004	1.21e+004
				(5.975)	(6.09)	(5.515)
$\lambda_{X,cl}$	0	0	0	15.4	13.9	14.6
				(2.054)	(2)	(2.139)
$\sigma_{JX,op}$	0	0	0	0.00234	0.00234	0.00219
				(15.59)	(15.82)	(13.79)
$\sigma_{JX,cl}$	0	0	0	0.0205	0.0214	0.0211
				(4.127)	(3.772)	(3.98)
μ_{JX}	0	0	0	0.000118	0.000116	8.53e-005
				(1.903)	(1.904)	(1.536)
	Measurement Error Parameters					
σ_Y	0	1e-010	5.73e-005	0	0.000284	0.000176
			(0.1058)		(2.332)	(0.8962)
λ_Y	0	0	0.0243	0	0	0.0121
			(4.339)			(1.528)
σ_{JY}	0	0	0.0053	0	0	0.00396
			(13.07)			(4.287)
Loglikelihood	11161.6	11161.6	11232.3	11373.1	11374.4	11380.5

The values inside parentheses are *t*-statistics.

M1 is the model without value jumps and without measurement errors.

M2 is the model without value jumps and with normally distributed measurement errors.

M3 is the model without value jumps and with heavy-tailed measurement errors.

M4-M6 correspond to M1-M3 except for allowing value jumps.

For M5, the estimate of σ_Y is constrained by the preset lower bound at 10^{-10}.

Since the standard *t*-statistics are no longer valid, they are not reported for this model.

Table 4: 2004 IBM stock prices, 1-hour sampling frequency

	M1	M2	M3	M4	M5	M6
	Asset Diffusion Parameters					
μ_X	-0.118	-0.11	-0.125	-0.29	-0.283	-0.271
	(-0.5469)	(-0.5276)	(-0.7788)	(-2.588)		(-2.374)
α_0	0.00808	0.00748	0.00874	0.00325	0.00329	0.00361
	(2.514)	(2.601)	(2.343)	(2.616)		(2.572)
α_1	0.914	0.912	0.899	0.871	0.871	0.857
	(26.98)	(27.11)	(20.89)	(19.14)		(16.3)
β_1	0.0452	0.0453	0.0592	0.074	0.0741	0.0766
	(2.51)	(2.473)	(2.423)	(3.714)		(3.632)
h_1	0.217	0.209	0.164	0.0649	0.0653	0.0643
	(4.029)	(3.83)	(3.584)	(1.688)		(1.405)
φ	0.415	0.425	0.378	0.578	0.58	0.568
	(32.76)	(26.17)	(29.44)	(9.394)		(7.752)
	Asset Jump Parameters					
$\lambda_{X,op}$	0	0	0	7.56e+003	7.62e+003	6.19e+003
				(4.559)		(3.863)
$\lambda_{X,cl}$	0	0	0	10.2	10.6	10.7
				(1.86)		(1.834)
$\sigma_{JX,op}$	0	0	0	0.00307	0.00306	0.0032
				(10.9)		(9.133)
$\sigma_{JX,cl}$	0	0	0	0.0242	0.0238	0.0237
				(2.907)		(2.942)
μ_{JX}	0	0	0	0.000196	0.000186	0.000205
				(1.804)		(1.571)
	Measurement Error Parameters					
σ_Y	0	0.000735	3.59e-005	0	1e-010	0.000222
		(3.429)	(0.02487)			(0.4505)
λ_Y	0	0	0.0378	0	0	0.0741
			(2.92)			(1.206)
σ_{JY}	0	0	0.00571	0	0	0.0027
			(9.584)			(2.769)
Loglikelihood	4898.5	4900.5	4928.8	5022.5	5022.5	5025.5

The values inside parentheses are *t*-statistics.

M1 is the model without value jumps and without measurement errors.

M2 is the model without value jumps and with normally distributed measurement errors.

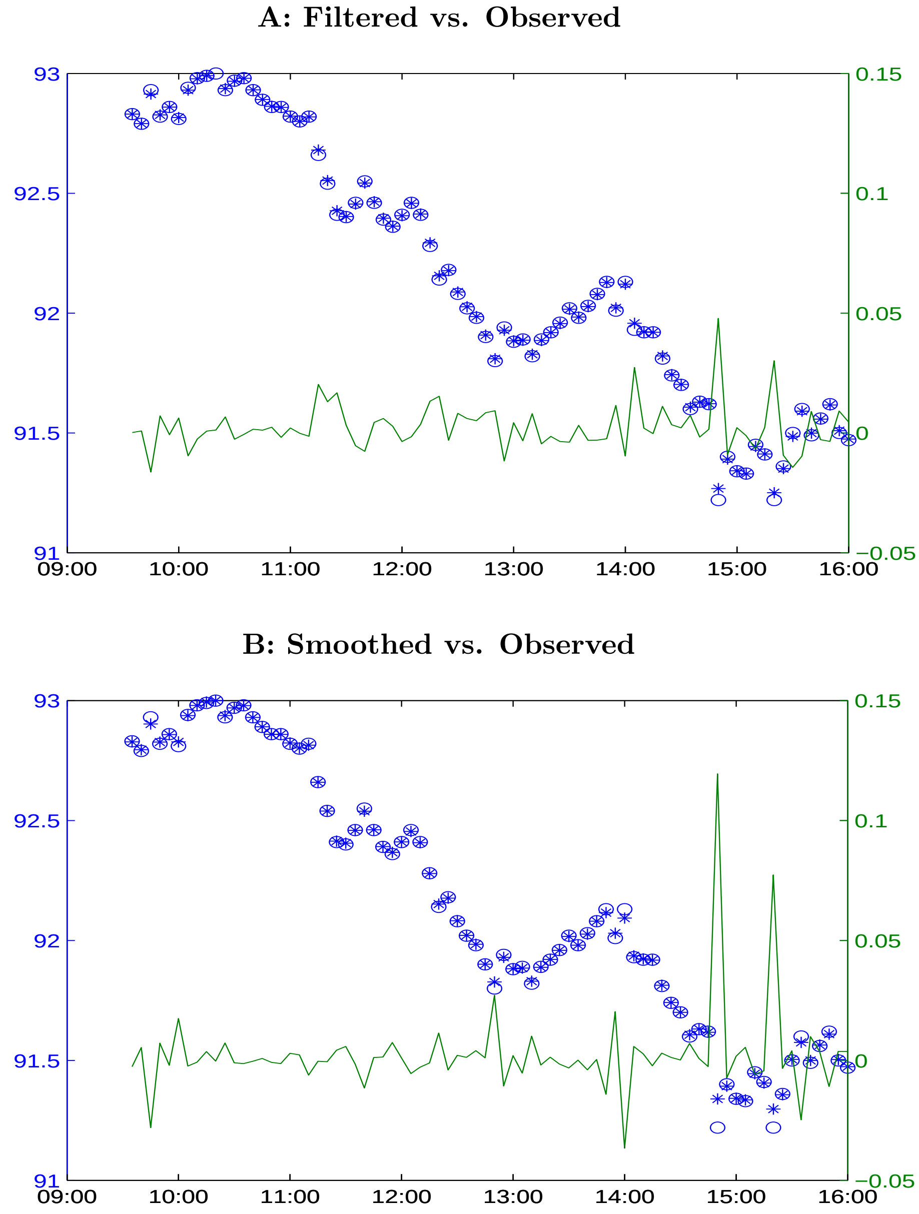
M3 is the model without value jumps and with heavy-tailed measurement errors.

M4-M6 correspond to M1-M3 except for allowing value jumps.

Table 5: IBM stock prices, multiple-year estimation results

	Estimates for $\lambda_{X,op}$			
	2002	2003	2004	2005
	5-minute sampling frequency			
No Noise	2.38e+004	2.66e+004	3.69e+004	2.85e+004
Normal Noise	2.32e+004	2.65e+004	3.63e+004	2.85e+004
Fat-tailed Noise	1.81e+004	1.98e+004	2.37e+004	1.74e+004
	10-minute sampling frequency			
No Noise	1.22e+004	1.27e+004	2.36e+004	1.37e+004
Normal Noise	1.23e+004	1.26e+004	2.02e+004	1.39e+004
Fat-tailed Noise	1.23e+004	1.15e+004	1.33e+004	1.02e+004
	30-minute sampling frequency			
No Noise	4.69e+003	4.87e+003	1.11e+004	3.36e+003
Normal Noise	4.47e+003	4.82e+003	1.12e+004	3.24e+003
Fat-tailed Noise	4.51e+003	5.17e+003	1.20e+004	2.92e+003
	1-hour sampling frequency			
No Noise	2.53e+003	2.96e+003	7.56e+003	4.09e+003
Normal Noise	2.53e+003	3.24e+003	7.61e+003	4.09e+003
Fat-tailed Noise	2.63e+003	2.95e+003	6.19e+003	3.73e+003
	Estimates for $\sigma_{JX,op}$			
	5-minute sampling frequency			
No Noise	0.00267	0.00165	0.00127	0.00139
Normal Noise	0.00269	0.00165	0.00127	0.00140
Fat-tailed Noise	0.00265	0.00166	0.00132	0.00151
	10-minute sampling frequency			
No Noise	0.00371	0.00222	0.00157	0.00198
Normal Noise	0.00370	0.00222	0.00165	0.00197
Fat-tailed Noise	0.00335	0.00214	0.00177	0.00202
	30-minute sampling frequency			
No Noise	0.00607	0.00380	0.00234	0.00370
Normal Noise	0.00617	0.00381	0.00233	0.00374
Fat-tailed Noise	0.00615	0.00362	0.00219	0.00378
	1-hour sampling frequency			
No Noise	0.00897	0.00524	0.00307	0.00417
Normal Noise	0.00873	0.00506	0.00305	0.00417
Fat-tailed Noise	0.00888	0.00518	0.00320	0.00421

Figure 1: Filtered, smoothed and observed stock prices for IBM on January 2, 2004 (5-minute frequency)

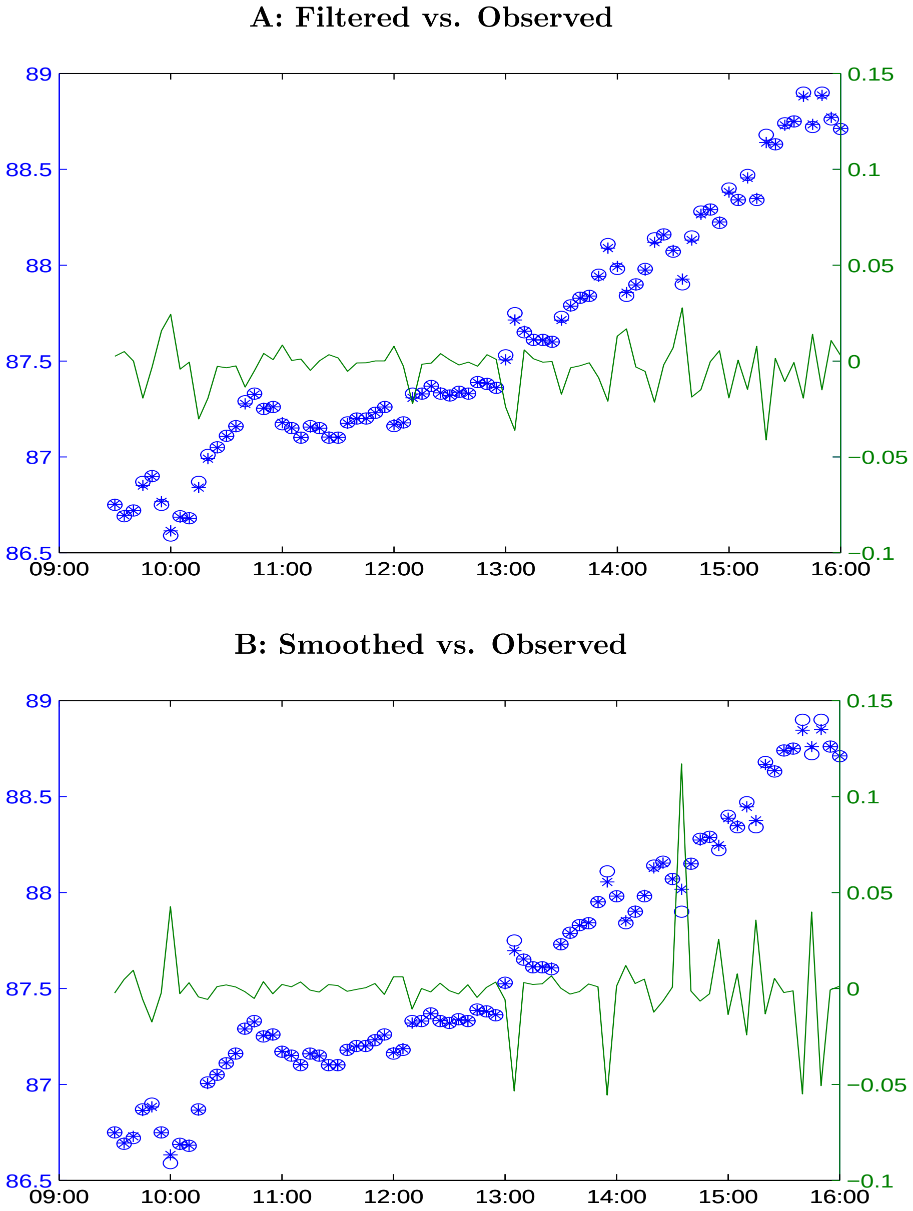


A: Filtered vs. Observed

B: Smoothed vs. Observed

For plot A, the left axis is for the filtered means of the "efficient" stock prices (∗) and the observed stock prices (o). Their differences are plotted against the right axis. The smoothed means vs. the observed stock prices are plotted in plot B in the same manner. The values are obtained by estimating the full model allowing for jumps in the stock price with heavy-tailed microstructure noises (i.e., M6).

Figure 2: Filtered, smoothed and observed stock prices for IBM on May 25, 2004 (5-minute frequency)



A: Filtered vs. Observed

B: Smoothed vs. Observed

For plot A, the left axis is for the filtered means of the "efficient" stock prices ($*$) and the observed stock prices (o). Their differences are plotted against the right axis. The smoothed means vs. the observed stock prices are plotted in plot B in the same manner. The values are obtained by estimating the full model allowing for jumps in the stock price with heavy-tailed microstructure noises (i.e., M6).