Microstructure Noise and the Dynamics of Volatility

Jin-Chuan Duan and Andras Fulop[*]

(First Draft: March 2010)

Abstract

To investigate the dynamic interrelationship between microstructure noise and price volatility, we devise a model under which the observed stock price is decomposed into a permanent component representing the efficient price and a transient microstructure noise. Both components are allowed to have time-varying volatilities through a GARCH-type specification, and the innovations in the two components can be non-Gaussian and will feed back to the volatility process. Bayesian techniques are used to estimate the resulting nonlinear and non-Gaussian state-space system. In particular, we develop a hierarchical sequential Monte Carlo method that samples from the posterior distribution of the fixed model parameters and that of the unobserved state variables. On the lower level and for a fixed set of model parameters, we run particle filters to obtain a point-wise estimate of the likelihood, which in essence integrates out the unobserved state variables in the likelihood. On the upper level, we obtain the posterior distribution using a sequential Monte Carlo sampler over the fixed parameters with the point-wise likelihood estimates. We show that the method works well with a finite number of particles. The heavy computational load is handled by a GPU-based parallel computer. Implementing the method on 100 firms from CRSP, we find that microstructure shocks have an important role in the volatility dynamics.

Keywords: Particle filter, Bayesian Methods, GARCH, Microstructure errors

[*] Duan is with Risk Management Institute and Department of Finance, National University of Singapore. E-mail: bizdjc@nus.edu.sg. Fulop is with ESSEC Business School, Paris. E-mail: fulop@essec.fr. Fulop is a research fellow at CREST.

1 Introduction

In this paper we ask whether transient changes in transaction prices have a role to play in the dynamics of volatility. To investigate this issue, we set up a model in which the observed transaction price is decomposed into a permanent component representing the efficient price and a transient microstructure noise. Both components are allowed to have fat tails and time-varying volatility. The volatility of microstructure noises is a constant multiple of the efficient price volatility. Our focus is the volatility equation, where in a GARCH-like specification, the conditional variance can depend on its own lagged value, on past efficient price innovations and past microstructure errors. The resulting system is a non-linear and non-gaussian state-space model.

We choose a likelihood-based Bayesian approach to inference over the joint distribution of the dynamic hidden states and the fixed model parameters. Particle filters provide the basic building block of our procedure. First, they allow us to obtain point-wise estimates of the likelihood for given model parameters. Second,they provide an approximation to the distribution of the hidden states conditional on the data and the model parameters. This suggests a marginalized sampling procedure where we first define a sampler over the fixed parameter using the point-wise likelihood estimates and then draw from the hidden states given the fixed parameters. However, to follow through this idea, the particle filter needs to be called tens of thousands of times at different fixed parameter sets, making the approach very time consuming. To make the computational task manageable, we implement a particle filter parallelized over the model parameters on a graphical processor achieving speedup factors of 40-100 over the CPU version. Then, to tackle our problem, we need a statistical method that (i) carries out the marginalized sampling routine (ii) can be parallelized over the model parameters. We achieve this objective by combining two recent statistical techniques. The first of these, the population MCMC (PMCMC) algorithm of Andrieu et al (2010) embeds a particle filter into an MCMC algorithm over the model parameters, satisfying our first criteria. Unfortunately it is sequential in the model parameters. To achieve parallelization, we turn to the Sequential Monte Carlo Sampler (SMCS) approach of Del Moral et al (2006). These authors propose to construct a sequence of bridging densities between the target distribution and an easy-to-sample initial density (the prior in our case). A set of N particle is passed through this sequence using Markov transition kernels and sequential monte carlo. The output is a population of N weighted samples representing the target.

We then define an SMCS procedure over the fixed parameters and use PMCMC kernels to move the particles towards the target. The procedure then consists of running N parallel PMCMC chains within an SMC algorithm over the model parameters. For each chain, computing the acceptance probabilities and the importance weights necessitates the point-wise likelihood estimates obtained from a particle filter using M particles. Hence our method is a hierarchical sequential monte carlo method. Applying the extended state-space construction of Andrieu et al (2010), we show that the method delivers samples from the joint distribution of the fixed model parameters and the hidden states for any finite M. In practice this allows us to use a moderate number for M as we do not need to resort to asymptotic arguments for validity.

We implement the methodology on daily equity data between 2002-2008 from CRSP on 100 randomly selected firms. Our main result is that for 89 firms in the sample, the data strongly favors the specification where microstructure errors have a role in the volatility dynamics (Model 2) as opposed to the basic model where only fundamental innovations enter the volatility equation.(Model 1) To understand our results better, we also estimate a stochastic volatility alternative where volatility is driven by fundamental noise and a volatility innovation independent of the fundamental noise and the microstructure errors.(Model 3) We find Model 3 to be observationally equivalent to Model 2. In other words, liquidity shocks and shocks to volatility seem to be closely intertwined. Once we allow for liquidity shocks to affect volatility we do not need independent volatility shocks. Model 1 is rejected by both other models for most firms.

We execute some checks to make sure that the quantities we call liquidity shocks are not pure artefacts of our methodology. First we find that the smoothed first two moments of the microstructure errors are consistent across different model specifications no matter whether the econometric identification comes from solely the autocorrelation structure of the observed return (Model 1) or it reflects both this autocorrelation structure and the volatility dynamics (Model 2). Second, we investigate whether the illiquidity proxy implied by our model, the standard deviation of the microstructure errors is related to common liquidity proxies in the cross-section. We find that firms with higher microstructure noise volatility tend to have bigger percentage bid ask spread, are smaller, have a smaller equity trading volume and have more zero return observations.

Our paper joins a growing literature utilizing sequential monte carlo techniques in the Bayesian estimation of financial and economic models. Fernandez-Villaverde and Rubio (2007) embed a particle filter within an MCMC algorithm and apply the algorithm to macroeconomics. Flury and Shepard (2009) show the usefulness of the PMCMC approach for an array of financial and economic models. Jasra et al (2008) estimate a stochastic volatility model using adaptive sequential monte carlo samplers. In an online setting, Carvalho et al (2007) provide methods to jointly estimate the model parameters and states in model that admit a sufficient statistic structure for the posterior. Johannes et al (2008) present an application to predictability and portfolio choice. The methodological contribution of our paper is to link sequential monte carlo samplers with the PMCMC methodology of Andrieu et al (2010). Our method inherits from PMCMC the feature that the proposals need to be designed only over the fixed parameters, instead of the joint space of the model parameters and the hidden states, which is typically much harder. The main advantages of working with sequential monte carlo samplers as opposed to a straight PMCMC procedure is that the method is easily parallelizable over the the model parameters.

We also build on the microstructure literature initiated by Roll (1984) that uses the moving-average structure of observed returns to estimate the magnitude of microstructure errors (effective spreads). Roll's Model is generalized by Hasbrouck (1993) to allow more general dependence structure of the noise and the efficient price, while in a recent contribution Bandi and Russell (2004) propose the use of high-frequency data to identify the volatility of microstructure errors in a non-parametric setting. We contribute to the empirical microstructure literature by setting up a parametric model that explicitly link microstructure errors to feed back to the volatility of the observed prices. Our main finding suggesting that stochastic volatility shocks may in large part be interpreted as liquidity shocks seems to add to our knowledge and may deserve further investigation.

2 Model

The efficient price, E_t follows a fat-tailed GARCH

$$E_{t+1} = \mu + E_t + \sigma_t \varepsilon_{t+1} \tag{1}$$

where the innovation has a generalized error distribution with unit variance. Its density is

$$\begin{aligned} f(\varepsilon) &= \frac{v \exp\left[-\frac{1}{2}\left|\frac{\varepsilon}{\lambda}\right|^{v}\right]}{\lambda 2^{(1+1/v)}\Gamma(1/v)} \\ \text{where } \lambda &= \left[2^{(-2/v)}\Gamma(1/v)/\Gamma(3/v)\right]^{1/2} \\ v &> 0 \end{aligned}$$

The measurement noise itself follows a GARCH process

$$Z_{t+1} = \delta \sigma_t \eta_{t+1} \tag{2}$$

Here we implicitly assume a constant signal-to-noise ratio δ and the measurement error η_{t+1} is $GED(v_2)$

The observation, Y_t is the sum of the efficient price and the measurement noise

$$Y_t = E_t + Z_t \tag{3}$$

To model the volatility process, we take an extension of the NGARCH specification:

$$\sigma_{t+1}^2 = \alpha_0(1-\alpha_1) + \alpha_1 \sigma_t^2 + \sigma_t^2 \left[\beta_1 g_{t+1} + \beta_2(\eta_{t+1}^2 - 1)\right] \tag{4}$$

g_{t+1} is the feedback from the fundamental innovation with asymmetry parameter γ, defined to have 0 mean:

$$g_{t+1} = (\varepsilon_{t+1} - \gamma)^2 - (1 + \gamma^2)$$

β_2 measures the effect of transient measurement noises on the volatility process and we allow for an independent normal volatility innovation, ζ_{t+1} through the parameter β_3. The persistence of the volatility process is α_1 while the unconditional variance is α_0.

To ensure positivity and stationarity of the variance process the following parameter restrictions are used:

$$\begin{aligned} \alpha_0, \alpha_1, \beta_1, \beta_2 &> 0 \qquad (5) \\ \alpha_1 - \beta_1(1+\gamma^2) - \beta_2 &> 0 \\ \alpha_1 &< 1 \end{aligned}$$

3 Estimation Method

Ours is a state-space model, where the hidden variables $X_t = (E_t, \sigma_t)$ completely determine the future evolution of the system. Denote the fixed model parameters by θ. The dynamics of the hidden states is determined by an initial density $X_1 \sim \mu_\theta(\cdot)$ and the transition density from (1) and (4)

$$X_{t+1} \mid (X_t = x) \sim f_\theta(\cdot|x) \tag{6}$$

The observations $y_t, t = 1, ..., T$ are linked to state of the system through the measurement equation (3) with the density denoted as

$$Y_t \mid (X_t = x) \sim g_\theta(\cdot|x) \tag{7}$$

For a given vector $(z_1, ..., z_t)$, use the notation $z_{1:t}$.

The objective is to perform Bayesian inference conditional on the observations $y_{1:T}$. If θ is a known parameter, the posterior is $p_\theta(x_{1:T} \mid y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T})$ where

$$p_\theta(x_{1:T}, y_{1:T}) = \mu_\theta(x_1) \prod_{n=2}^{T} f_\theta(x_n|x_{n-1}) \prod_{n=1}^{T} g_\theta(y_n \mid x_n) \tag{8}$$

If θ is unknown and $p(\theta)$ is the prior distribution over θ the joint posterior is

$$p(\theta, x_{1:T} \mid y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T}) p(\theta) \tag{9}$$

In the following we first assume that θ is known and present particle filtering methods that give approximations to $p_\theta(x_{1:T} \mid y_{1:T})$ and the marginal likelihood $p_\theta(y_{1:T})$. Then, we present a method to sample from the joint distribution $p_\theta(\theta, x_{1:T} \mid y_{1:T})$.

3.1 Particle Filter for fixed θ

Particle filters provide a sequential approximation to the densities $p_\theta(x_{1:t} \mid y_{1:t})$ and the marginal likelihoods $p_\theta(y_{1:t})$ for $t = 1, ..., T$ and a fixed θ by propagating a set of particles through the system. In particular, assume that at time $t-1$ we have a set of particles $X^k_{1:t-1}, k = 1, ..., M$ with attached weights W^k_{t-1} whose empirical distribution approximates $p_\theta(x_{1:t-1} \mid y_{1:t-1})$

$$\hat{p}_\theta(x_{1:t-1} \mid y_{1:t-1}) := \sum_{k=1}^{M} W^k_{t-1} \delta_{X^k_{1:t-1}}(x_{1:t-1}) \tag{10}$$

Then, the identity

$$p_\theta(x_{1:t} \mid y_{1:t}) \propto p_\theta(x_{1:t-1} \mid y_{1:t-1}) f_\theta(x_t|x_{t-1}) g_\theta(y_t|x_t)$$

suggests sampling particles from $\hat{p}_\theta(x_{1:t-1} \mid y_{1:t-1})$ and extending them to $x_{1:t}$ using importance sampling and some proposal distribution $q_\theta(\cdot|y_t, x_{t-1})$. The importance weights that compensate for the difference between the target and the proposal are $s_t(x_t) = \frac{f_\theta(x_t|x_{t-1})g_\theta(y_t|x_t)}{q_\theta(x_t|y_t,x_{t-1})}$. This will produce a weighted sample approximately distributed according to $p_\theta(x_{1:t} \mid y_{1:t})$

Use the notation $W_t := (W_t^1, ..., W_t^M)$ for the normalized importance weights at time t and $\mathcal{F}(\cdot|\mathbf{p})$ for the discrete probability distribution on $1, \ldots, M)$ with probabilities $\mathbf{p} = (p_1, \ldots, p_M)$. Then, the particle filtering algorithm is as follows

Initialization At $t = 1$

1. Sample $X_1^k \sim q_\theta(\cdot|y_1)$
2. Compute and normalize the weights

$$\begin{aligned} \tilde{w}_1(X_1^k) &= \frac{\mu_\theta(X_1^k) g_\theta(y_1|X_1^k)}{q_\theta(X_1^k|y_1)} \\ W_1^k &= \frac{\tilde{w}_1(X_1^k)}{\sum_{i=1}^M \tilde{w}_1(X_1^i)} \end{aligned}$$

Recursions At $t = 2, \ldots, T$

1. Sample $A_{t-1}^k \sim \mathcal{F}(\cdot|\mathbf{W}_{n-1})$
2. Sample $X_t^k \sim q_\theta(\cdot|y_t, X_{t-1}^{A_{t-1}^k})$ and set $X_{1:t}^k = (X_{1:t-1}^{A_{t-1}^k}, X_t^k)$
3. Compute and normalize the weights

$$\begin{aligned} \tilde{w}_t(X_{1:t}^k) &= \frac{f_\theta(X_t^k|X_{t-1}^{A_{t-1}^k}) g_\theta(y_1|X_t^k)}{q_\theta(X_t^k|y_t, X_{t-1}^{A_{t-1}^k})} \\ W_t^k &= \frac{\tilde{w}_t(X_{1:t}^k)}{\sum_{i=1}^M \tilde{w}_t(X_{1:t}^i)} \end{aligned}$$

This algorithm provides an estimate of the joint posterior density of the hidden states

$$\hat{p}_\theta(x_{1:T}|y_{1:T}) = \sum_{k=1}^{M} W_T^k \delta_{X_{1:T}^k}(x_{1:T}) \tag{11}$$

which can be sampled by sampling from the discrete distribution $\mathcal{F}(\cdot|\mathbf{W}_T)$ and taking the corresponding particle $X_{1:T}^k$. Further, we also obtain an estimate of the marginal likelihood given by

$$\begin{aligned} \hat{p}_\theta(y_{1:T}) &= \hat{p}_\theta(y_1)\prod_{t=2}^{T}\hat{p}_\theta(y_t|y_{1:t-1}) \\ \hat{p}_\theta(y_t|y_{1:t-1}) &= \frac{1}{M}\sum_{k=1}^{M}\tilde{w}_t(X_{1:t}^k) \end{aligned} \tag{12}$$

Above, A_{t-1}^k represents the parent at time $t-1$ of particle $X_{1:t}^k$. Using the notation $\mathbf{A}_n = (A_n^1, \dots, A_n^M)$, the joint density of the parents chosen at n is

$$r(\mathbf{A}_{n-1}|W_{n-1}) = \prod_{k=1}^{M}\mathcal{F}(A_{n-1}^k|\mathbf{W}_{n-1})$$

Further, define B_t^k to be the index that the ancestor of particle $X_{1:T}^k$ had at time t. These can recursively be obtained as follows from A_t^k:

- At $t = T$ we have $B_t^k = k$
- At $t < T$ we have $B_t^k = A_t^{B_{t+1}^k}$

Then we have the identity $X_{1:T}^k = (X_1^{B_1^k}, \dots, X_T^{B_T^k})$ and $B_{1:T}^k = (B_1^k, \dots, B_T^k = k)$ is the ancestral lineage of particle k. For a general introduction to particle filters see Doucet and Gordon (2001) while for theoretical results see Del Moral (2004).

3.2 Sampling over θ

In the previous subsection we have seen that for given fixed parameters θ, the particle filtering algorithm provides estimates of the hidden dynamic states $\hat{p}_\theta(x_{1:T}|y_{1:T})$ and the likelihood of the data after marginalizing out the hidden states, $\hat{p}_\theta(y_{1:T})$. This suggests that a sensible approach to obtain the joint posterior of the dynamic states and the model parameters $p(\theta, x_{1:T} \mid y_{1:T})$ is to first sample from the marginal distribution of θ and then to draw from the hidden states conditional on the fixed parameters. To operationalize this, we first need a sampler over the model parameters, θ utilizing the point-wise likelihood estimates $\hat{p}_\theta(y_{1:T})$. Then, estimates of the hidden states given the model parameters could be obtained by drawing from the particles produced by these particle filters. This approach is attractive as instead of designing a sampler over the joint distribution of $(\theta, x_{1:T})$ which can be very high-dimensional when the data sample size T is large, we only need to design samplers over θ, a much easier task.

In any such sampling routine we typically need to call the particle filtering algorithm tens of thousands of times, leading to a substantial computational load. To make our approach manageable, we have programmed a parallelized particle filter for our model on a graphical processing unit (GPU) based NVIDIA Tesla processor containing 240 processing cores. Lee et. al (2010) have already demonstrated the usefulness of GPU-based processors to dramatically speed up sequential monte carlo routines when the number of particles gets very large. Our implementation is somewhat different, as in our model we do not need a large number of particles, M for satisfactory performance for a given model parameter set, θ. To benefit from the computational power of the GPU, we parallelize the routine in the dimension of the model parameters. In essence each thread runs a particle filter for a given θ. Table 1 shows timing results on the speedup we have achieved using this approach compared to a CPU-based implementation. In the experiment, the number of data points is set to $T = 1500$ while the number of particles in the point-wise particle filters is $M = 100$. When the routine is called for 1024 sets of model parameters, which is the number we use in the applications later, the GPU-based implementation achieves a 40× speedup over the CPU. When the number of model parameters gets even higher the speed up achieves 100×. In short, our GPU-based implementation removes the computational bottleneck if we can find a statistical algorithm that is parallel in θ.

Thus, what we need is a statistical method that (i) implements the marginalization idea by directly sampling from the model parameters using estimates of the point-wise likelihood estimates and (ii) is parallelizable in θ. In the following we describe an approach that satisfies these two criteria. To achieve our goal, we combine two recent statistical innovations. The first of these, the particle MCMC (PMCMC) approach of Andrieu et. al (2010) embed a particle filter into an MCMC algorithm over θ. Crucially, Andrieu et al. (2010) show that for any finite number of particles M, the equilibrium distribution of the chain is the joint posterior $p(\theta, x_{1:T} \mid y_{1:T})$, i.e.

the estimation error in the point-wise likelihoods does not invalidate the algorithm. While the PMCMC approach satisfies our first requirement, being an ingenuous implementation of the marginalization idea, unfortunately it is inherently sequential in θ. To achieve parallelization over θ, we turn to the sequential monte carlo sampler (SMCS) approach of Del Moral et al (2006). In this method, one defines a sequence of bridging densities between an easy-to-sample distribution (the prior) and the ultimate target (the posterior) and propagates a sample of particles through these densities. The particles are moved using MCMC kernels and appropriate weights are attached to them to make sure that the population at the end represents the target distribution. We combine the two methods by running an SMCS routine over N sets of fixed parameters θ and moving the particles around using a variant of PMCMC kernels. The combined method will satisfy both of our requirements. First, it is trivially parallelizable over θ, a property it inherits from SMCS. Second, it is a valid implementation of the marginalization idea for any finite number of particles in the pointwise likelihood evaluations, a feature borrowed from the PMCMC framework. In the following, we briefly present the two building blocks of our approach, PMCMC and SMCS and then describe the hierarchical sequential monte carlo routine.

3.2.1 Particle MCMC

This section introduces the PMCMC sampler from Andrieu et al. (2010). We have seen in section 3.1 that for fixed θ, the particle filter provides estimates of the marginal likelihood $p_\theta(y_{1:T})$ and the conditional density $p_\theta(x_{1:T}|y_{1:T})$. If we are interested in sampling from the joint distribution $p(\theta, x_{1:T}|y_{1:T})$, the decomposition $p(\theta, x_{1:T}|y_{1:T}) = p(\theta|x_{1:T})p_\theta(x_{1:T}|y_{1:T})$ suggests the following "ideal" marginalized MH update:

$$q\{(\theta^*, x^*_{1:T})|(\theta, x_{1:T})\} = q(\theta^*|\theta)p_{\theta^*}(x^*_{1:T}|y_{1:T})$$

One only needs to choose the proposal over θ, while sampling from the hidden states is perfectly adapted. The resulting MH acceptance ratio is

$$\frac{p(\theta^*, x^*_{1:T}|y_{1:T})}{p(\theta, x_{1:T}|y_{1:T})}\frac{q\{(\theta, x_{1:T})|(\theta^*, x^*_{1:T})\}}{q\{(\theta^*, x^*_{1:T})|(\theta, x_{1:T})\}} = \frac{p_{\theta^*}(y_{1:T})p(\theta^*)}{p_\theta(y_{1:T})p(\theta)}\frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}$$

While this "ideal" MH cannot be implemented, Andrieu et al (2010) suggest to run an analogue using the approximations from the particle filter, $\hat{p}_\theta(y_{1:T})$ and $\hat{p}_\theta(x_{1:T}|y_{1:T})$. Note that an early appearance of the same idea can be found in

Fernandez-Villaverde and Rubio-Ramirez (2007). The resulting particle marginal MH algorithm is

Initialization $i = 0$

- Set $\theta(0)$ arbitrarily
- Run a particle filter at $\theta(0)$, sample $X_{1:T}(0) \sim \hat{p}_{\theta(0)}(x_{1:T}|y_{1:T})$ and let $\hat{p}_{\theta(0)}(y_{1:T})$ denote the marginal likelihood estimate

Iterations $i > 0$

- Sample $\theta^* \sim q(\cdot|\theta(i-1))$
- Run a particle filter at θ^*, sample $X^*_{1:T} \sim \hat{p}_{\theta^*}(x_{1:T}|y_{1:T})$ and let $\hat{p}_{\theta^*}(y_{1:T})$ denote the marginal likelihood estimate. Note, that the random numbers used in the particle filter need to be independent across the iterations.
- The MH acceptance probability is

  $$\alpha(i) = 1 \wedge \frac{\hat{p}_{\theta^*}(y_{1:T})p(\theta^*)}{\hat{p}_{\theta(i-1)}(y_{1:T})p(\theta(i-1))} \frac{q(\theta(i-1)|\theta^*)}{q(\theta^*|\theta(-1))}$$

  We then have

  $$\begin{aligned} \theta(i) = \theta^*, X_{1:T}(i) = X^*_{1:T}, \hat{p}_{\theta(i)}(y_{1:T}) = \hat{p}_{\theta^*}(y_{1:T}) & \quad \text{with probability } \alpha(i) \\ \theta(i) = \theta(i-1), X_{1:T}(i) = X_{1:T}(i-1), \hat{p}_{\theta(i)}(y_{1:T}) = \hat{p}_{\theta(i-1)}(y_{1:T}) & \quad \text{with probability } 1-\alpha(i) \end{aligned}$$

The key result of Andrieu et al (2010) (their Theorem 4) consists of showing that this markov chain admits $p(\theta, x_{1:T}|y_{1:T})$ as its stationary distribution. In other words, the estimation error in $\hat{p}_\theta(y_{1:T})$ and $\hat{p}_\theta(x_{1:T}|y_{1:T})$ does not change the equilibrium distribution of the chain[1]. The PMCMC algorithm implements the marginalization idea, but it is unfortunately sequential in θ. In the following we describe a method that will allow us to achieve parallelization.

[1] It will however influence the dependence and hence the efficiency of the chain

3.2.2 Sequential Monte Carlo Samplers

Let us assume for the purposes of this subsection that we can actually evaluate the likelihood of the parameters θ, $p_\theta(y_{1:T})$ and ignore the hidden states for the moment. Then, our task consists of sampling from the posterior $p(\theta|y_{1:T}) \propto p_\theta(y_{1:T})p(\theta)$ using a method that is parallel in θ. Sequential Monte Carlo Samplers (SMCS), introduced by Del Moral et al (2006) provide a methodology to do this. The main idea of the method is to begin with an easy-to-sample distribution and traverse through a sequence of densities to the ultimate target which is much harder to sample from. In particular construct the following sequence of densities:

$$\begin{aligned} \pi_l(\theta) &= \frac{\gamma_l(\theta)}{Z_l}, l = 1, \ldots, P \\ \gamma_l(\theta) &= p_\theta(y_{1:T})^{\xi_l} p(\theta) \text{ where } \xi_1 = 0, \ldots, \xi_P = 1 \end{aligned}$$

$\pi_1(\theta) = p(\theta)$ is the prior, while the final distribution $\pi_P(\theta) = p(\theta, x_{1:T}|y_{1:T})$ is the posterior. Further, the ratio of the normalizing constants $\frac{Z_P}{Z_1}$ is the marginal likelihood of the model, important for Bayesian model choice. Then, SMCS suggests to first sample N points using some importance sampler $\eta_1(\theta)$ targeting $\gamma_1(\theta)$ and then to recursively sample from $\gamma_l(\theta)$ by moving these points using some Markov kernel $K_l(\theta, \theta')$ for $l = 2, \ldots, P$. In essence, the method gradually gathers information on the target and adapts the sampler accordingly. This is implemented by using the location of the previously sampled particles in the new proposals. The change in the tempering parameter, $\xi_l - \xi_{l-1}$ determines the speed of learning and can be tuned depending on the difficulty of the task.

One of the contributions of Del Moral et al (2006) lies in proposing an auxiliary target density that facilitates the computation of importance weights. In particular, introduce the following auxiliary targets

$$\begin{aligned} \tilde{\pi}_l(\theta_{1:l}) &= \frac{\tilde{\gamma}_l(\theta_{1:l})}{Z_l} \\ \tilde{\gamma}_l(\theta_{1:l}) &= \gamma_l(\theta_l) \prod_{k=1}^{l-1} L_k(\theta_{k+1}, \theta_k) \qquad (13) \end{aligned}$$

where $L_{k-1}(\theta_k, \theta_{k-1})$ is an arbitrary backward Markov kernel. Then, $\tilde{\pi}_l(\theta_{1:l})$ admits $\pi_l(\theta_l)$ as a marginal by construction and the two distributions share the same normalizing constant Z_l. The main advantage of working with $\tilde{\pi}_l(\theta_{1:l})$ is that it facilitates the computation of the importance weights. Define the importance function $\eta_l(\theta_{1:l})$ as the joint distribution of the initial importance sampler $\eta_1(\theta)$ and the Markov Kernels $K_l(\theta, \theta')$

$$\eta_l(\theta_{1:l}) = \eta_1(\theta_1) \prod_{k=1}^{l-1} K_{k+1}(\theta_{k-1}, \theta_k)$$

Then, applying importance sampling to the auxiliary target, we obtain the unnormalized importance weights

$$\begin{aligned} s_l(\theta_{1:l}) &= \tilde{\gamma}_l(\theta_{1:l})/\eta_l(\theta_{1:l}) \\ &= s_{l-1}(\theta_{1:l-1})\tilde{s}_l(\theta_{l-1}, \theta_l) \end{aligned}$$

where the incremental weights are

$$\tilde{s}_l(\theta_{l-1}, \theta_l) = \frac{\gamma_l(\theta_l)L_{l-1}(\theta_l, \theta_{l-1})}{\gamma_{l-1}(\theta_{l-1}} K_l(\theta_{l-1}, \theta_l) \tag{14}$$

In the method, there is a great degree of freedom in the choice of the forward markov kernels $K_l(\theta_{l-1}, \theta_l)$ and of the artificial backward ones, $L_{l-1}(\theta_l, \theta_{l-1})$. Here let us assume that the forward kernel $K_l(\theta_{l-1}, \theta_l)$ is an MCMC kernel leaving $\pi_l(\theta_l)$ invariant. Then, we can define the backward kernel as follows

$$L_{l-1}(\theta_l, \theta_{l-1}) = \frac{\pi_l(\theta_{l-1})K_l(\theta_{l-1}, \theta_l)}{\pi_l(\theta_l)} \tag{15}$$

and the incremental weights become

$$\tilde{s}_l(\theta_{l-1}, \theta_l) = \frac{\gamma_l(\theta_{l-1})}{\gamma_{l-1}(\theta_{l-1})} \tag{16}$$

In our problem the incremental importance weights takes the particularly simple form of

$$\tilde{s}_l(\theta_{l-1}, \theta_l) = p_{\theta_{l-1}}(y_{1:T})^{\xi_l - \xi_{l-1}}$$

The algorithm so far has been already proposed in prior work by Neal (2001). The key innovation of Del Moral et al (2006) compared to Neal (2001) is to apply resampling ideas to the problem. As the algorithm proceeds the weights $s_l(\theta_{1:l})$ tend to become concentrated to a few particles, a phenomenon known as particle degeneracy. To counterweight this effect, Del Moral et al (2006) propose to resample the particle population whenever the diversity of the particles falls below a certain value. One measure of degeneracy of the particle set is the effective sample size (ESS) criterion: $ESS = \frac{1}{\sum_{j=1}^{N}\left(S_l^{(j)}\right)^2}$ where $S_l^{(j)}$ is the normalized weight of particle j in step l. In general the ESS is between 1 and N. Then, one should resample if the ESS drops below a pre-specified value, e.g. $N/2$. This yields the following SMC algorithm over θ

Initialization At $l = 1$

- Sample $\Theta_1^{(i)} \sim \eta_1(\cdot)$
- Attach importance weights

$$s_1(\Theta_1^{(i)}) = \frac{\gamma_1(\Theta_1^{(i)})}{\eta_1(\Theta_1^{(i)})}$$

  and normalize the weights

$$S_1^{(i)} = \frac{s_1(\Theta_1^{(i)})}{\sum_{j=1}^{N} s_1(\Theta_1^{(j)})}$$

Recursions At $l = 2, \ldots, P$

- If $ESS < N/2$, resample and set $S_{l-1}^{(i)} = \frac{1}{N}$
- Sample $\Theta_l^{(i)} \sim K_l(\Theta_{l-1}^{(i)}, \cdot)$
- Compute incremental weights

$$\tilde{s}_l(\Theta_{l-1}^{(i)}, \Theta_l^{(i)}) = p_{\Theta_{l-1}^{(i)}}(y_{1:T})^{\xi_l - \xi_{l-1}}$$

  and normalize the weights

$$S_l^{(i)} = \frac{\tilde{s}_l(\Theta_{l-1}^{(i)}, \Theta_l^{(i)}) S_{l-1}^{(i)}}{\sum_{j=1}^{N} \tilde{s}_l(\Theta_{l-1}^{(j)}, \Theta_l^{(j)}) S_{l-1}^{(j)}}$$

We can estimate the ratio of the subsequent normalizing constants as

$$\frac{\widehat{Z_l}}{Z_{l-1}} = \sum_{j=1}^{N} S_{l-1}^{(j)} \tilde{s}_l(\Theta_{l-1}^{(i)}, \Theta_l^{(i)})$$

The ratio Z_P/Z_1 can be estimated as

$$\frac{\widehat{Z_P}}{Z_1} = \prod_{l=2}^{P} \frac{\widehat{Z_l}}{Z_{l-1}}$$

One can interpret this algorithm as running N parallel interacting MCMC chains where importance weights are used to allow for the fact that the previous state of the chain has not been the target distribution.

3.2.3 Hierarchical Sequential Monte Carlo

Now we can propose a method that (i) samples from the joint distribution of $(x_{1:T}, \theta)$ using the marginalization idea of PMCMC (ii) can be parallelized in θ following SMCS. Basically, we embed PMCMC kernels within an SMCS algorithm over θ. We call the method Hierarchical Sequential Monte Carlo, as it consists of running a sequential monte carlo routine over θ, while it uses point-wise particle filters given θ to implement the Markov kernels, to compute the importance weights in the SMCS algorithm and to sample from the hidden states $x_{1:T}$.

In the previous discussion on SMCS, we have assumed that the likelihood function $p_\theta(y_{1:T})$ can be evaluated and we ignored the sampling over the hidden states. While in our problem we do not have access to $p_\theta(y_{1:T})$, we can run a point-wise particle filter for θ and obtain an estimate $\hat{p}_\theta(y_{1:T})$. This can be used to estimate the incremental weights in (3.2.2). Further, the argument of Andrieu et al (2010) suggests that we can use the following Markov Kernels $K_l(\theta_{l-1}, \theta_l)$ targeting $\hat{p}_\theta(y_{1:T})^{\xi_l} p(\theta)$ to move the particles around:

- Sample the fixed parameters according to a random walk move $\theta^* \sim h_{l-1}(\cdot \mid \theta_{l-1})$. Here $h_{l-1}(\cdot \mid \theta_{l-1})$ can be adaptive, i.e. it can depend on the past of the particles.

- Run a particle filter at θ^*, compute the estimated normalizing constant $\hat{p}^* = \hat{p}_{\theta^*}(y_{1:T})$

- Sample a particle from $X_{1:T}^* \sim \hat{p}_{\theta^*}(x_{1:T}|y_{1:T})$

- Accept the proposal $(\theta^*, \hat{p}^*, X^*_{1:T})$ with probability

$$1 \wedge \frac{(\hat{p}^*)^{\xi_l} p(\theta^*)}{\hat{p}_{l-1}^{\xi_l} p(\theta_{l-1})} \frac{h_{l-1}(\theta_{l-1} \mid \theta^*)}{h_{l-1}(\theta^* \mid \theta_{l-1})}$$

For simplicity, assume that we can directly sample from the prior distribution. Then, the Hierarchical Sequential Monte Carlo Algorithm with P steps and N particles in the SMC scheme over θ and M particles in the point-wise particle filters will take the following form:

Initialization At $l = 1$

For $i = 1, \dots, N$

1. Sample $\Theta^{(i)}(1) \sim p(\theta)$ from the prior distribution over the fixed parameters
2. Run a particle filter for each $\Theta^{(i)}(1)$ with M particles and compute the estimate of the normalizing constant $\hat{p}^{(i)}(1) = \hat{p}_{\Theta^{(i)}(1)}(y_{1:T})$. Note that the rnadom numbers used in the particle filter are assumed to be independent across the samples and across iteration. Sample a particle from $\hat{p}_{\Theta^{(i)}(1)}(x_{1:T}|y_{1:T})$, denote it as $X^{(i)}_{1:T}(1)$.
3. Attribute equal weights $S^{(i)}(1) = 1/N$ to the particles

Recursions At $l = 2, \dots, P$

1. If $ESS < N/2$, resample and set $S^{(i)}(l-1) = \frac{1}{N}$
2. Compute the incremental weights

$$\tilde{s}^{(i)}(l) = \hat{p}^{(i)}(l-1)^{\xi_l - \xi_{l-1}}$$

estimate the ratio of normalizing constants

$$\frac{\widehat{Z_l}}{Z_{l-1}} = \sum_{i=1}^{N} S^{(i)}(l-1)\tilde{s}^{(i)}(l)$$

and normalize the weights

$$S^{(i)}(l) = \frac{\tilde{s}^{(i)}(l) S^{(i)}(l-1)}{\sum_{j=1}^{N} \tilde{s}^{(j)}(l) S^{(j)}(l-1)}$$

3. Move the particles using the PMCMC kernel

   - Draw fixed parameters from $\Theta^{(i,*)}(l) \sim h_l(\cdot|\Theta^{(i)}(l-1))$
   - Run a particle filter for each $\Theta^{(i,*)}(l)$ with M particles and compute the estimate of the normalizing constant $\hat{p}^{(i,*)}(l) = \hat{p}_{\Theta^{(i,*)}(l)}(y_{1:T})$. Sample a particle from $\hat{p}_{\Theta^{(i,*)}(l)}(x_{1:T}|y_{1:T})$, denote it as $X_{1:T}^{(i,*)}(l)$.
   - The acceptance probability is

$$\alpha_l^{(i)} = 1 \wedge \frac{\left(\hat{p}^{(i,*)}(l)\right)^{\xi_l} p\left(\Theta^{(i,*)}(l)\right)}{\left(\hat{p}^{(i)}(l-1)\right)^{\xi_l} p\left(\Theta^{(i)}(l-1)\right)} \frac{h_l\left(\Theta^{(i)}(l-1) \mid \Theta^{(i,*)}(l)\right)}{h_l\left(\Theta^{(i,*)}(l) \mid \Theta^{(i)}(l-1)\right)}$$

   Then, we have that

$$\begin{aligned} \left(\Theta^{(i)}(l), \hat{p}^{(i)}(l), X_{1:T}^{(i)}(l)\right) &= \left(\Theta^{(i,*)}(l), \hat{p}^{(i,*)}(l), X_{1:T}^{(i,*)}(l)\right) \text{ with prob. } \alpha_l^{(i)} \\ \left(\Theta^{(i)}(l), \hat{p}^{(i)}(l), X_{1:T}^{(i)}(l)\right) &= \left(\Theta^{(i)}(l-1), \hat{p}^{(i)}(l-1), X_{1:T}^{(i)}(l-1)\right) \text{ otherwise} \end{aligned}$$

After running the algorithm, the empirical distribution of the weighted sample $\{S^{(i)}(P), \Theta^{(i)}(P), X_{1:T}^{(i)}(P)\}, i = 1, \ldots, N$ provides a sample from the joint distribution of $p(\theta, x_{1:T}|y_{1:T})$ for any finite number of M for a large enough N. The validity of the algorithm is shown in the Appendix and uses the extended state space construction of Andrieu et. al (2010) in the SMCS context.

4 Empirical Results

4.1 Data, Priors and Simulation Parameters

We have estimated the model on daily equity price data from CRSP between 2002-2008 on 100 randomly selected firms. We have excluded penny stocks (prices lower than 2 USD) and we only used price data originating from actual transactions. Further, we have restricted our investigation to firms traded either on NYSE or on NASDAQ and names where we have at least 1500 observations in the sample period.

We have specified independent priors for the model parameters and obtained the prior distribution by specifying their 5^{th} and 95^{th} percentiles. These quantities and the chosen density function for each parameter can be found in Table 2. As can be seen from the Table, we have tried to use relatively non-informative priors. In addition, we enforce the stationarity and non-negativity restrictions from (5) in the prior.

In the particle filter given fixed parameters we have used a gaussian importance function which incorporates new information in a gaussian approximation of the model. This seemed to be a consistently efficient choice. Throughout we have used a relatively low number of particles $M = 100$. For each firm, the variance has been initialized at the variance of the observed returns, while the efficient price process at the first observed price observations. To make sure that the initialization does not drive the results, the first estimated likelihood of the first 10 observations have been dropped from the estimation routine.

In the SMC sampler over the fixed parameters, we have used $N = 1024$ particles and $P = 100$ steps in the SMC algorithm. Following advice from Jasra et al (2007), we have chosen to focus more of the computational effort on the initial steps. In particular, the tempering parameter ξ_l was uniformly spaced between $[0, 0.15]$ in the first 20 steps, between $[0.15, 0.4]$ between steps 20-60 and finally the last 40 steps were equally distributed between $[0.4, 1]$. The estimation of one model specification for one firm took roughly 7 minutes on a workstation using one Nvidia Tesla card.

In the MCMC kernels, we used a normal random walk for parameters without nonnegativity restrictions (i.e. for μ and γ). For all the other parameters we used a lognormal random walk. We allowed the MCMC kernel to depend on the past of the particles. In particular, the variance of the random walk increments at step l was set to $c_l \times Var_{l-1}$ where Var_{l-1} is the empirical variance of the target at step $l - 1$. The scaling constant c_l is chosen to keep the average acceptance probability in the MCMC step between 0.2 and 0.4. In particular if the acceptance probability in step $l - 1$ falls below 0.2, $c_l = c_{l-1}/1.5$ If in contrast it grows above 0.4 we set $c_l = c_{l-1} \times 1.5$. The scale adaptation has been stopped at the 50th iteration, otherwise the algorithm tended to decrease the scale without limits. The reason for this phenomenon is that the Monte Carlo noise from using a finite sample of particles M in the point-wise likelihood evaluations tends to decrease the acceptance probabilities below the range obtained by the ideal algorithm using an infinite M.

4.2 Parameter Estimates

Table 3 reports cross-sectional statistics across the 100 firms on the posterior means of the model parameters for two model specifications. In Model 1 we set $\beta_2 = 0$ so the only innovation to the volatility dynamics comes from fundamental noises. In Model 2 β_2 is estimated so volatility can be affected by both fundamental innovations and microstructure noises.

There are several patterns emerging from the Table. First, comparing estimates from Model 1 and Model 2 one can see that microstructure errors seem to important predictors of volatility. In fact when one allows feedback from microstructure errors to volatility, one observes that the coefficient on the microstructure error (β_2) is twice as large on average as the coefficient on the fundamental innovations (β_1). Another interesting feature across the two models is that while the average value of volatility persistence (α_1) is similar, the lower tail seems to be significantly lower for Model 1. In other words, for some firms, allowing microstructure errors seems to lead to a higher estimated volatility persistence. To understand these results in further detail, Table 3 reports the cross-sectional correlation matrix of the parameter estimates for Model 2. The most interesting number here is the high negative correlation (-0.75) between the volatility persistence parameter, α_1 and the feedback parameter on microstructure noise, β_2, suggesting that firms with large noise have lower volatility persistence. In fact these are exactly the firms where allowing for feedback to volatility from microstructure noise leads to a decrease in the estimated volatility persistence.

In both specifications, the average value of the signal-to-noise ratio is $\delta \approx 0.2$, showing that in our sample transient pricing errors are non-negligeable, their magnitude is roughly one-fifth of daily fundamental price changes. There are no surprises on the behavior of the fundamental noise ε_t. For the average firm it is fat-tailed with the GED degree of freedom parameter close to one and the innovations typically feeds back to volatility with a leverage parameter $\gamma > 0$. In contrast, interestingly, the data does not seem to contain much information on the shape of the measurement noise distribution v_2, the length of the posterior central 90% credible interval is very close to its prior analogue.

To formally compare the two specifications we have also computed the Bayes factor between them. In particular, if the marginal likelihood of Model 2 is Z_2 while that of Model B is Z_1, then the relative likelihood of Model 2 compared to Model 1 is the Bayes factor defined as $B_{21} = \frac{Z_2}{Z_1}$. Kass and Raftery (1995) propose the following scale to interpret the evidence in favour of Model 2 compared to Model 1

2 ln(B_{AB})	B$_{AB}$	Evidence in favour of Model 2
0 to 2	1 to 3	Not worth more than a bare mention
2 to 6	3 to 20	Positive
6 to 10	20 to 150	Strong
10 <	150 <	Very Strong

Reinforcing the need for microstructure noises in predicting the volatility process, out of the 100 firms, there is strong evidence in the data for Model 2 compared to Model 1 for 89 firms.

5 Robustness Checks

5.1 Stochastic Volatility

We wanted to check how our model relate to a standard stochastic volatility model. In order to do so we have modified the variance equation to the following

$$\sigma_{t+1}^2 = \alpha_0(1-\alpha_1) + \alpha_1\sigma_t^2 + \sigma_t^2\left[\beta_1 g_{t+1} + \beta_3(\zeta_{t+1}^2 - 1)\right] \tag{17}$$

where we allow for an independent normal volatility innovation, ζ_{t+1} through the parameter β_3. We have chosen the prior over β_3 to follow a beta distribution with its central 90% probability mass covering $[0.01, 0.3]$. Let us denote this specification as Model 3.

Panel A of Table 5 reports the parameters estimates from Model 3, while Panel B reports the number of firms where the data contains strong evidence for the model in the row compared to the model in the column. The main message from the table is that the independent stochastic volatility noise seems to play the same role in the volatility process as the microstructure error we used before. Looking at Panel A, Model 3 seems to behave identically to our Model 2 before. In fact, the cross-sectional correlation coefficient between estimates of β_2 in Model 2 and estimates of β_3 in Model 3 is 0.93. Further, looking at the evidence from the pairwise Bayes Factors in Panel B, we see that for most firms the data favors Model 2 and Model 3 to Model 1, but it does not discriminate between Model 2 and Model 3.

A question that comes up is whether the measurement noises implied by our methodology is a robust feature of the data instead of being a simple artefact of a given model specification. To address this issue we have computed the average correlation between the smoothed estimates of the measurement noises $E(Z_t|y_{1:T})$ in the three model specifications in Panel C. We can see that the average correlation between these quantities is very high, around 90% suggesting that the measurement errors are robustly pinned down by the observed data. In particular, the microstructure errors in Model 1 and Model 2 are very similar. This is a good news as it suggests that in Model 2 microstructure errors are not simply rotated to play the role of some unrelated volatility variable. To investigate this point in greater detail we have also computed the average correlations between the squared microstructure errors $E(Z_t^2|y_{1:T})$ in Model 1 and Model 2 and the independent volatility innovations

$E(\sigma_{t-1}^2\zeta_t^2|y_{1:T})$ in Model 3. Remember that in Model 1, microstructure noises are exclusively pinned down by the autocorrelation structure of the observed returns while in Model 2 they are identified by both the autocorrelation structure and by the volatility process. In spite of the different source of identification, $E(Z_t^2|y_{1:T})$ is closely related across the two models with a correlation parameter of 80%. Further, the squared microstructure error from Model 1 has a correlation of 65% with the independent volatility noise from Model 3, showing that when there are shocks in volatility, the scale of measurement noises tend to be higher. Overall these results tends to suggest that stochastic volatility shocks and liquidity shocks are closely related and one are observationally equivalent. One we account for the latter, we do not need the former to explain the data.

5.2 Relationship of the estimates to liquidity proxies

Last, we would like to relate a measure of liquidity from our model to commonly used liquidity proxies to make sure that our results are reasonable. A natural measure of illiquidity in our setup is the standard deviation of the microstructure noise, $\delta E(\sigma_t|y_{1:T})$.For each firm, we have computed the time-series average of this variable and then investigated whether it is related to some liquidity proxies in the cross-section using the Spearman rank correlation. The liquidity proxies we investigated were the average percentage bid-ask spread, the average log size, the average log volume of shares traded and the percentage of zero returns. Table 6 reports that our illiquidity measure tends to be larger for firms with higher bid-ask spreads, lower size, smaller trading volume and more zero returns. All the relationships have y=the sign we expect a priori and the only one which is not significant is the relationship to the log volume. We have also investigated the time series Spearman correlation of the percentage bid-ask spreads for each firm with $E(\sigma_t|y_{1:T})$ and average correlation number was 0.24. In sum, our measure seems to be related to these commonly used liquidity measures.

6 Conclusions

In this paper we investigate what role microstructure errors play in the volatility dynamics of equity prices. We set up a model where observed prices are decomposed into permanent fundamental innovations and transient microstructure effects. The volatility of the prices follow a GARCH-like process where both fundamental innovations and microstructure errors can affect the volatility process. To estimate the model, we propose a novel Bayesian technique using a hierarchical sequential monte carlo algorithm and tackle the computational load using a GPU-based computer.

We estimate the model on daily equity data on CRSP for 100 firms between 2002-2008. our main finding is that microstructure errors play an important role in volatility. For most firms the data strongly favors the model that allows a feedback from microstructure errors to volatility compared to the version where this channel is shut down. Further, when we compare the model with microstructure errors in the volatility process with a stochastic volatility model we find the two specifications observationally equivalent. Shocks to volatility that are orthogonal to fundamental innovations seem closely related to liquidity shocks from our microstructure model. In other word, once we allow both fundamental innovations and microstructure noises to affect volatility we do not need extra stochastic volatility shocks to explain the data. Further, the microstructure noises seem robust across different model specifications and their volatility is related to common liquidity proxies in the cross-section.

There are several research directions that these results open up. First, it would be interesting to dig deeper and try to understand the economic drivers of why microstructure errors affect volatility. To empirically test the different potential causes we can proceed in the tradition of GARCH models and introduce the terms proxying for the various economic forces into the volatility equation. For example, Brunnermeier and Pedersen (2009) describe a mechanism where microstructure innovations may feed back on the fundamental price process through the effect of margins. In particular, if margins are set using a Var-type risk measure based on observed returns and there is volatility clustering in fundamentals, then a liquidity shock will lead to an increase in margins. In their model this in turn will lead to subsequent price changes due to liquidity spirals. To empirically test whether this mechanism contributes to our findings, we should obtain a time-varying proxy of funding liquidity and estimate a specification where the feedback from microstructure innovations to volatility is allowed to depend on the level of asset liquidity. Second, the econometric methodology proposed, using hierarchical sequential monte carlo, is very general and can be applied to any model that can be written in a state-space form. Further, using the GPU implementation it is fast enough to handle large data-sets. In particular, the methodology seems well suited to estimate rich structural empirical microstructure models or parametric models of high-frequency data that contain jumps, stochastic volatility and microstructure effects.

Appendix: Validity of the Hierarchical Sequential Monte Carlo algo-

rithm

This Appendix contains the proof that the algorithm described in Section 3.2.3 delivers samples from the joint distribution of $p(\theta, x_{1:T}|y_{1:T})$. The proof is basically an application of the extended target density construction from Andrieu et al (2010). The main idea here is to define an extended target density that includes all the random variables produced by the particle filtering algorithm and which admits the $p(\theta, x_{1:T}|y_{1:T})$ as a marginal.

In the proof that follow we use the notation introduced in Section 3.1. In addition use the notation $\bar{\mathbf{X}}_t = (X_t^1, \ldots, X_t^M)$. Then, it is easy to check that the joint density of the random variables produced by a point-wise particle filter run at θ, $\big(\bar{\mathbf{X}}_1, \ldots, \bar{\mathbf{X}}_T, \mathbf{A}_1, \ldots, \mathbf{A}_{T-1}\big)$ is

$$\psi_\theta\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) = \left\{\prod_{i=1}^{M} q_\theta(x_1^i|y_1)\right\} \prod_{t=2}^{T} \left\{ r_\theta(\mathbf{a}_{t-1}|\mathbf{w}_{t-1}) \prod_{i=1}^{M} q_\theta(x_t^i|y_1, x_{t-1}^{a_{t-1}^i}) \right\} \tag{18}$$

If we further denote by K the index of the particle that is sampled from $\hat{p}_\theta(x_{1:T}|y_{1:T})$, the joint density of $\big(K, \bar{\mathbf{X}}_1, \ldots, \bar{\mathbf{X}}_T, \mathbf{A}_1, \ldots, \mathbf{A}_{T-1}\big)$ is

$$w_T^k \psi_\theta\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right)$$

Now consider the following artificial extended target experiment devised by Andrieu et al (2010) over $\big(\Theta, K, \bar{\mathbf{X}}_1, \ldots, \bar{\mathbf{X}}_T, \mathbf{A}_1, \ldots, \mathbf{A}_{T-1}\big)$

- Choose the particle index at T and its ancestral lineage from a uniform distribution on $(1, \ldots, M)^T$
  - Sample K from a uniform over $(1, \ldots, M)$. Set $B_T^K = K$
  - For each $t = T-1, ..., 1$ sample $A_t^{B_{t+1}}$ from a uniform $(1, \ldots, M)$ and set $B_t^K = A_t^{B_{t+1}}$
- Sample $(\Theta, X_1^{B_1^K}, \ldots, X_1^{B_T^K})$ from the target distribution $p(\theta, x_{1:T}|y_{1:T})$. (Of course this is only a hypothetical sampling experiment, we do not know how to do this) By definition, the k^{th} particle path $X_{1:T}^K = X(_1^{B_1^K}, \ldots, X_1^{B_T^K})$.
- Run a conditional particle filtering algorithm compatible with $(B_1^K, \ldots, B_T^K)$ and $(\Theta, X_1^{B_1^K}, \ldots, X_1^{B_T^K})$.

The density function of the extended target is

$$
\begin{aligned}
& p^M\left(\theta, k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) \\
= \;& \frac{1}{N^T} \times p(\theta, x_{1:T}^k | y_{1:T}) \times \frac{\psi_\theta\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right)}{q_1(x_1^{b_1^k}|y_1) \prod_{t=2}^T r(b_{t-1}^k|\mathbf{w}_{t-1}) q_t(x_t^{b_t^k}|x_{t-1}^{b_{t-1}^k}, y_t)} \\
= \;& w_T^k \psi_\theta\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) \hat{p}_\theta(y_{1:T}) p(\theta) \qquad (19)
\end{aligned}
$$

By construction, the marginal of $(\Theta, X_1^{B_1^K}, \ldots, X_1^{B_T^K})$ we obtain from this auxiliary target will be distributed according to our original target $p(\theta, x_{1:T}|y_{1:T})$.

Now we define an SMCS routine over the extended space $\left(\Theta, K, \bar{\mathbf{X}}_1, \ldots, \bar{\mathbf{X}}_T, \mathbf{A}_1, \ldots, \mathbf{A}_{T-1}\right)$ using the following sequence of densities:

$$
\begin{aligned}
& \gamma_l\left(\theta, k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) \\
= \;& w_T^k \psi_\theta\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) \hat{p}_\theta(y_{1:T})^{\xi_l} p(\theta) \qquad (20)
\end{aligned}
$$

where $\xi_1 = 0, \ldots, \xi_P = 1$. Then the marginal distribution of $\gamma_1(\cdot)$ is exactly the prior, while $\gamma_P(\cdot)$ is the extended target in (19) with a marginal $p(\theta, x_{1:T}|y_{1:T})$.

Further, the proposals in the PMCMC kernels proposed in Section 3.2.3 will have a density

$$
\begin{aligned}
& q_l^M\left(\theta, k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) \\
= \;& w_T^k(\theta) \psi_\theta\left(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right) h_{l-1}(\theta \mid \theta_{l-1}) \qquad (21)
\end{aligned}
$$

where the likelihood ratio in the MH sampler is

$$
\begin{aligned}
& \frac{\gamma_l\left(\theta, k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right)}{q_l^M\left(\theta, k, \bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_T, \mathbf{a}_1, \ldots, \mathbf{a}_{T-1}\right)} \\
= \;& \frac{\hat{p}_\theta(y_{1:T})^{\xi_l} p(\theta)}{h_{l-1}(\theta \mid \theta_{l-1})}
\end{aligned}
$$

leading to the acceptance probabilities described in the text. Last, the importance weights that need to be attached are

$$
\frac{\gamma_l(U(l-1))}{\gamma_{l-1}(U(l-1))} = \hat{p}_{\Theta_{l-1}}(y_{1:T})^{\xi_l - \xi_{l-1}}
$$

Here for simplicity we have denoted by $U(l)$ all the random variables defined on the extended space in iteration l.

References

[1] Andrieu C, Doucet A, Holenstein A (2010) Particle markov chain monte carlo. Journal of Royal Statistical Society B 72:1-33

[2] Bandi FM, Russell JR (2004) Full-information transaction costs. Working paper

[3] Brunnermeier M, Pedersen L (2009) Market Liquidity and Funding Liquidity. Review of Financial Studies, 22:2201-2238

[4] Cappe O, Douc R, Gullin A, Marin JM, Robert CP (2008) Adaptive importance sampling in general mixture classes. forthcoming Statistics and Computing 18:447-459

[5] Fernandez-Villaverde J, Rubio JF (2007) Estimating Macroeconomic Models: A Likelihood Approach. Review of Economic Studies 74:1059-1087

[6] Carvalho CM, Johannes M, Lopes HF,Polson N (2007) Particle Learning and Smoothing. Working Paper

[7] Del Moral P (2004) Feynman-Kac formulae genealogical and interacting particle systems with applications. Springer, New York

[8] Del Moral P, Doucet A, Jasra A (2006) Sequential Monte Carlo samplers. Journal of Royal Statistical Society B 68:411:436

[9] Doucet A, De Freitas N, Gordon N (eds) (2001) Sequential Monte Carlo Methods in Practice. Springer-Verlag

[10] Flury T, and Shephard N (2008) Bayesian inference based only on simulated likelihood: Particle filter analysis of dynamic economic models. Technical report, Nuffield College, Oxford University

[11] Hasbrouck J (1993) Assessing the quality of a security market: a new approach to transaction cost measurement. Review of Financial Studies 6:191-212

[12] Johannes M, Korteweg AG, Polson N (2008) Sequential learning, predictive regressions, and optimal portfolio returns. Working Paper

[13] Kass RE, Raftery AE (1995) Bayes Factors. Journal of the American Statistical Association 90:773-795

Table 1: Computation time of particle filters on GPU vs CPU

N	Time on GPU	Time on CPU	CPU/GPU
32	2.45	3.74	1.5
64	2.39	7.47	3.1
128	2.41	14.93	6.2
256	2.46	29.85	12.1
512	2.62	59.76	22.8
1024	2.88	119.48	41.5
2048	3.3	238.91	72.4
4096	4.4	477.82	108.6
8192	10	955.64	95.6
16384	18.63	1911.28	102.6

This Table report the computation time in seconds of the particle filter on a CPU and on GPU for N sets of fixed parameters. The number of data points is T=1500, the number of particle M=100. The CPU is a Intel Xeon E5420 2.5 GHZ while the GPU is an Nvidia Tesla C1060 with 240 cores.

[14] Neal RM (2001) Annealed Importance Sampling. Statistics and Computing 11:125-139

[15] Jasra A, Stephens DA, Doucet A, Tsagaris T (2008) Inference for Levy Driven Stochastic Volatility Models Via Adaptive Sequential Monte Carlo. Technical report

[16] Lee A, Yau C, Giles MB, Doucet A, Holmes CC (2008) On the utility of graphics cards to perform massively parallel simulation with advanced Monte Carlo methods. Technical report, Oxford University

[17] Roll R (1984) A simple measure of the effective bid-ask spread in an efficient market. Journal of Finance 39:1127-1139.

Table 2: Prior Specification

	Density Function	5th Percentile	95th Percentile
μ	Normal	- 1/252	1/252
α_0	Inv. Gamma	Var_i/5	Var_i*5
α_1	Beta	0.3	0.95
β_1	Beta	0.01	0.03
β_2	Beta	0.01	0.03
γ	Normal	-3	3
v	Inv. Gamma	1	3
v_2	Inv. Gamma	1	3
δ	Gamma	0.01	1

This table reports the prior specification over the fixed parameters. The priors over the individual parameters are independent. Var_i is the variance of the observed return for the firm

Table 3: Posterior Means of Model Parameters

	Model 1			**Model 2**		
	Mean	5th Percentile	95th Percentile	Mean	5th Percentile	95th Percentile
μ	0.0001	-0.0011	0.0011	0.0001	-0.0010	0.0011
α_0	0.0007	0.0002	0.0017	0.0008	0.0002	0.0019
α_1	0.94	0.82	0.99	0.95	0.89	0.99
β_1	0.10	0.04	0.21	0.04	0.01	0.09
β_2				0.15	0.06	0.31
γ	0.52	0.00	1.20	0.94	-0.21	2.03
v	1.05	0.82	1.42	1.19	0.92	1.56
v_2	1.90	1.50	2.38	1.94	1.53	2.34
δ	0.20	0.04	0.47	0.20	0.05	0.46

This table reports cross-sectional statistics across the 100 firms on the posterior means of the parameters for different model specifications.

Model 1: Only fundamental noise in volatility equation
Model 2: Fundamental noise+Microstructure noise in volatility equation

Table 4: Cross-sectional Correlation Matrix of Parameters

	μ	α_0	α_1	β_1	β_2	γ	v	v_2	δ
μ	1.00								
α_0	-0.38	1.00							
α_1	0.04	-0.27	1.00						
β_1	0.24	-0.10	-0.11	1.00					
β_2	-0.11	0.35	-0.75	-0.04	1.00				
γ	-0.01	-0.11	0.18	-0.47	-0.14	1.00			
v	0.34	-0.03	0.01	-0.04	-0.20	0.09	1.00		
v_2	-0.01	0.05	0.10	-0.05	0.01	0.02	0.15	1.00	
δ	-0.16	-0.23	0.04	0.06	-0.22	-0.03	-0.40	-0.32	1.00

This table reports the cross-sectional correlation of the parameter estimates for the 100 firms for Model 2.

Model 2: Fundamental noise+Microstructure noise in volatility equation

Table 5

Panel A: Parameter Estimates for Model 3

	Mean	5th Percentile	95th Percentile
μ	0.0002	-0.0011	0.0011
α_0	0.0008	0.0002	0.0019
α_1	0.95	0.88	0.99
β_1	0.04	0.01	0.10
β_3	0.16	0.06	0.32
γ	0.89	-0.47	1.86
v	1.21	0.94	1.58
v_2	1.91	1.45	2.45
δ	0.19	0.04	0.47

Panel A reports cross-sectional statistics across the 100 firms on the posterior means of the parameters

Panel B: Bayes Factors showing strong evidence (BF>20)

	Model 1	Model 2	Model 3
Model 1		0	0
Model 2	89		4
Model 3	88	2	

In each cell, Panel B reports the number of firms where the data favors the Model in the row compared to the Model in the column as measured by the Bayes Factors.

Panel C: Correlations of some smoothed quantities

	$E(Z_t^2)$ From M1	$E(Z_t^2)$ From M2	$E(\sigma^2_{t-1}\xi_t^2)$ From M3
$E(Z_t^2)$ From M1	1		
$E(Z_t^2)$ From M2	0.80	1.00	
$E(\sigma^2_{t-1}\xi_t^2)$ From M3	0.65	0.83	1

	$E(Z_t)$ From M1	$E(Z_t)$ From M2	$E(Z_t)$ From M3
$E(Z_t)$ From M1	1		
$E(Z_t)$ From M2	0.88	1.00	
$E(Z_t)$ From M3	0.93	0.93	1

Panel C reports the average correlations between some smoothed quantities across different model specifications

Table 6

	Spearman rho	pvalue
% Bid-ask Spread	0.24	0.02
Log(Size)	-0.31	0.00
Log(Volume)	-0.14	0.17
% ZeroRet	0.32	0.00

This table computes the cross-sectional Spearman rank correlation coefficient between the time-series average of the microstructure noise stdev. $\delta\sigma_t$ and different liquidity proxies