Forecasting observables in state space models: does the choice of filter matter?*

Patrick Leung[†], Catherine S. Forbes[‡], Gael M. Martin[§]and Brendan McCabe[¶]

December 2, 2020

Abstract

We investigate the impact of filter choice on forecast accuracy in state space models. The filters are used both to estimate the posterior distribution of the parameters, via a particle marginal Metropolis-Hastings (PMMH) algorithm, and to produce draws from the filtered distribution of the final state. Multiple filters are entertained, including two new data-driven methods. Simulation exercises are used to document the performance of each PMMH algorithm, in terms of computation time and the efficiency of the chain. We then produce the forecast distributions for the one-step-ahead value of the observed variable, using a fixed number of particles and Markov chain draws. Despite distinct differences in efficiency, the filters yield virtually identical forecasting accuracy, with this result holding under both correct and incorrect specification of the model. This invariance of forecast performance to the specification of the filter also characterizes an empirical analysis of S&P500 daily returns.

KEYWORDS: Bayesian Prediction; Particle MCMC; Non-Gaussian Time Series; State Space Models; Unbiased Likelihood Estimation; Sequential Monte Carlo.

JEL Classifications: C11, C22, C58.

1 Introduction

Markov chain Monte Carlo (MCMC) schemes for state space models in which the likelihood function is estimated using a particle filter, have expanded the toolkit of the Bayesian statistician. The seminal work on *particle* MCMC (PMCMC) by Andrieu, Doucet and Holenstein (2010) in fact illustrates the more general concept of *pseudo-marginal* MCMC, in which insertion of an unbiased estimator of the likelihood within a Metropolis-Hastings (MH) algorithm is shown to yield the correct invariant distribution (Beaumont, 2003; Andrieu and Roberts, 2009). Subsequent work on PMCMC by Flury and Shephard (2011) and Pitt, dos Santos Silva, Giordani and Kohn (2012) has explored the interface between different filtering-based estimates of the likelihood and the mixing properties of the resultant PMCMC algorithms in a variety of settings, with the features of the true data generating process - in particular the signal-to-noise ratio in the assumed state space model (SSM) - playing a key role in

[*]The research has been supported by Australian Research Council Discovery Grants DP150101728 and DP170100729. All empirical data used in the paper will be made available, as per the requirements of the journal.

[†]Department of Econometrics and Business Statistics, Monash University.

[‡]Department of Econometrics and Business Statistics, Monash University.

[§]Department of Econometrics and Business Statistics, Monash University. Corresponding author; email: gael.martin@monash.edu; postal address: 20 Chancellors Walk, Monash University, Clayton 3800, Australia.

[¶]Management School, University of Liverpool.

the analysis. See Whiteley and Lee (2014), Del Moral, Doucet, Jasra, Lee, Yau and Zhang (2015), Del Moral and Murray (2015), Guarniero, Johansen and Lee (2017), Deligiannidis, Doucet and Pitt (2018), Doucet and Lee (2018) and Quiroz, Tran, Villani and Kohn (2018, 2019) for a range of more recent contributions to the area.

The aim of this paper is a very particular one: to explore the implications for *forecast accuracy* of using different particle marginal MH (PMMH) algorithms. That is, we address the question of whether or not the specific nature of the filter - used both to construct the likelihood estimate and to draw from the filtered distribution of the final state - affects the forecast distribution of the observed, or measured, random variable in the state space model. This is, as far as we are aware, a matter that has not yet been investigated, and is one of practical relevance to those researchers whose primary goal is prediction, rather than inference *per se*. In *theory*, all filters that we consider - providing unbiased estimators of the likelihood function as they do - will produce exact inference on the static parameters and the latent states, given a large enough number of simulation draws. Hence, one would anticipate little difference in the estimated forecast distribution for the observable variable, as long as each filter is optimized in the requisite way. In *practice*, for any given number of simulated draws, different performances will be observed for any given number of simulated draws. We are interested in ascertaining whether such differences - which may well yield conflicting posterior *inferences* - matter when it comes to prediction.[1]

The performance of several well-established filters is explored, namely the bootstrap particle filter (BPF) of Gordon, Salmond and Smith (1993), the auxiliary particle filter (APF) of Pitt and Shephard (1999) and the unscented particle filter (UPF) of van de Merwe, Doucet, de Freitas and Wan (2000). In order to broaden the scope of the investigation, we introduce and include two new particle filters. Drawing on a particular representation of the components in an SSM, as first highlighted in Ng, Forbes, Martin and McCabe (2013), our first new filter provides a mechanism for generating independent proposal draws using information on the current data point only, and we use the term 'data-driven particle filter' (DPF) to refer to it as a consequence. The second filter is a modification of this basic DPF - a so-called 'unscented' DPF (UDPF) - which exploits unscented transformations (Julier, Uhlmann and Durrant-Whyte, 1995; Julier and Uhlmann, 1997) in conjunction with the DPF mechanism to produce draws that are informed by both the current observation and the previous state. The paper thus includes a range of particle filters that, in varying ways, allow for differential impact of the current observation and the past (forecasted) information about the current state variable. Hence, the conclusions we draw about forecast performance cannot be deemed to be unduly influenced by focusing on too narrow a class of filter.

Using simulation, from a range of state space models, and under different scenarios for the signal-

[1] Whilst the focus here is on (potentially) exact Bayesian algorithms, as opposed to algorithms that explicitly produce an approximate posterior, this work still bears some resemblance to other work that explores forecasting outcomes in approximate Bayesian settings (e.g. Quiroz, Nott and Kohn, 2018; Frazier, Maneesoonthorn, Martin and McCabe, 2019).

to-noise ratio, we first document the performance of each filter-specific PMMH algorithm, in terms of computation time and the efficiency of the chain. We then produce the forecast distributions for the one-step-ahead value of the observed random variable, documenting forecast performance over a hold-out period. The key result is that, despite differences in efficiency, the alternative filters yield virtually identical forecasting accuracy, with this result holding under both correct and incorrect specification of the true DGP.

To further substantiate this conclusion, we use alternative filters in an empirical setting in which a relatively simple stochastic volatility model is estimated using data from Standard and Poor's Composite Index, denoted simply as the S&P 500. Given that this data has typically been modelled using a much more complex process, it is likely that the model is misspecified, with inference itself being impacted by that misspecification. Despite this fact we find, again, that the probabilistic forecasts produced by the different filters are virtually identical and, hence, yield the same degree of forecast accuracy.

In Section 2.1 we begin with an outline of the role played by particle filtering in likelihood estimation and implementation of a PMMH scheme followed, in Section 2.2, with a description of its role in producing an estimate of the one-step-ahead forecast distribution for the observed variable. In Section 3 we first provide a brief outline of the existing, and well-known, filters that we include in our investigations: the BPF, APF and UPF. The two new filters that we introduce, the DPF and the UDPF, are then described. The computational benefit of using a multiple matching of particles (Lin, Zhang, Cheng and Chen, 2005) in the production of the likelihood estimate is explored in the context of the DPF, and in Appendix A.3 it is established that the likelihood estimators resulting from both new filters are unbiased.

Two different simulation exercises are conducted in Section 4. The first investigates the relative computational burden of each of several distinct filter types, along with the resulting impact on the mixing properties of the corresponding Markov chain. This exercise is based on three alternative state space models: i) the linear Gaussian model; ii) the stochastic conditional duration model of Bauwens and Veredas (2004) (see also Strickland, Forbes and Martin, 2006); and iii) the stochastic volatility model of Taylor (1982) (see also Shephard, 2005). The alternative filter types are used to estimate the likelihood function within an adaptive random walk-based MH algorithm. For each method we record both the 'likelihood computing time' associated with each filtering method - namely the average time taken to produce a likelihood estimate with a given level of precision at some representative (vector) parameter value - and the inefficiency factors associated with the resultant PMMH algorithm. In so doing we follow the spirit of the exercise undertaken in Pitt *et al.* (2012), amongst others, in which a balance is achieved between computational burden and the efficiency of the resultant Markov chain; measuring as we do the time taken to produce a likelihood estimate that is sufficiently accurate to yield an acceptable mixing rate in the chain.

In the second simulation exercise, forecasting performance is the focus. We estimate the one-step-ahead forecast (or predictive) distribution of the observed variable, for each of the different filters and associated PMMH algorithms, repeating the exercise (using expanding windows) over a hold-out period, and assessing forecast accuracy using a logarithmic scoring rule (Gneiting and Raftery, 2007). We first generate artificial data from a stochastic volatility model, and produce the forecast distributions using the correctly specified model. We then keep the forecast model the same, but generate data from a stochastic volatility model with both price and volatility jumps, in order to assess the impact on forecast performance of model misspecification. To allow the documented differences in the performance of the different filters to potentially affect forecast accuracy, in this exercise we hold the number of particles, and the number of MCMC draws, fixed for all PMMH methods. Despite this, the alternative filters yield virtually identical forecasting accuracy, with this result holding under both correct and incorrect specification of the true DGP. In Section 5, an empirical study is undertaken, in which the competing PMMH algorithms are used to produce one-step-ahead forecast distributions from a stochastic volatility model for the S&P 500 data. Again, despite restricting the different filters to operate with the same number of particles, and the resulting Markov chains to have the same number of iterations, the resulting forecast performance of the competing methods is essentially equivalent. Section 6 concludes.

2 PMMH and Forecasting

2.1 Unbiased likelihood estimation and PMMH

In our context, an SSM describes the evolution of a latent state variable, denoted by x_t, over discrete times $t = 1, 2, ...$, according to the state transition probability density function (pdf),

$$p\left(x_{t+1}|x_t, \theta\right), \tag{1}$$

and with initial state probability given by $p\left(x_0|\theta\right)$, where θ denotes a vector of unknown parameters. The observation in period t, denoted by y_t, is modelled conditionally given the contemporaneously indexed state variable via the conditional measurement density

$$p\left(y_t|x_t, \theta\right). \tag{2}$$

Without loss of generality we assume that both x_t and y_t are scalar.

Typically, the complexity of the model is such that the likelihood function,

$$L(\theta) = p(y_{1:T}|\theta) = p(y_1|\theta)\prod_{t=2}^{T} p(y_t|y_{1:t-1}, \theta), \tag{3}$$

where $y_{1:t-1} = (y_1, y_2, \ldots, y_{t-1})'$, is unavailable in closed form. Particle filtering algorithms play a role here by producing (weighted) draws from the filtering density at time t, $p(x_t|y_{1:t}, \theta)$, with those draws in turn being used, via standard calculations, to estimate the one-step ahead predictive densities of which the likelihood function in (3) is comprised. The filtering literature is characterized by different methods of producing and weighting the filtered draws, or particles, with importance sampling principles being invoked, and additional MCMC steps also playing a role in some cases. Not surprisingly, performance of the alternative algorithms (often measured in terms of the accuracy with which the filtered density itself is estimated) has been shown to be strongly influenced by the empirical characteristics of the SSM, with motivation for the development of a data-driven filter coming from the poor performance of the BPF (in particular) in cases where the signal-to-noise ratio is large; see Giordani, Pitt and Kohn (2011) and Creal (2012) for extensive surveys and discussion, and Del Moral and Murray (2015) for a more recent contribution.

A key insight of Andrieu *et al.* (2010) is that particle filtering can be used to produce an unbiased estimator of the likelihood function which, when embedded within a suitable MCMC algorithm, yields exact Bayesian inference, in the sense that the invariant distribution of the Markov chain is the posterior of interest, $p(\theta|y_{1:T})$. In brief, by defining u as the vector containing the canonical identically and independently distributed (*i.i.d.*) random variables that underlie a given filtering algorithm, and defining the corresponding filtering-based estimate of $L(\theta)$ by $\widehat{p}_u(y_{1:T}|\theta) = p(y_{1:T}|\theta, u)$, the role played by the auxiliary variable u in the production of the estimate is made explicit. Andrieu *et al.* demonstrate that under the condition that

$$E_u[\widehat{p}_u(y_{1:T}|\theta)] = p(y_{1:T}|\theta), \tag{4}$$

i.e., that $\widehat{p}_u(y_{1:T}|\theta)$ is an unbiased (and non-negative) estimator of the likelihood function, then the marginal posterior associated with the joint distribution,

$$p(\theta, u|y_{1:T}) \propto p(y_{1:T}|\theta, u) \times p(\theta) \times p\left(u\right), \tag{5}$$

is $p(\theta|y_{1:T})$. Hence, this marginal posterior density can be accessed via an MH algorithm for example, in which the estimated likelihood function, $\widehat{p}_u(y_{1:T}|\theta)$, replaces the exact (but unavailable) $p(y_{1:T}|\theta)$ in the ratio that defines the MH acceptance probability at iteration i in the chain,

$$\alpha = \min\left\{1, \frac{\widehat{p}\left(y_{1:T}|\theta^c\right) p\left(\theta^c\right) q\left(\theta^{(i-1)}|\theta^c\right)}{\widehat{p}\left(y_{1:T}|\theta^{(i-1)}\right) p\left(\theta^{(i-1)}\right) q\left(\theta^c|\theta^{(i-1)}\right)}\right\},$$

where $q(\cdot)$ denotes the candidate density, θ^c a candidate draw from $q(\cdot)$, and $\theta^{(i-1)}$ the value of the chain at iteration $i-1$. Such an algorithm is referred to a particle *marginal* MH (PMMH) algorithm since draws from the augmented joint in (5) represent draws from the desired marginal, $p(\theta|y_{1:T})$.

Flury and Shephard (2011) subsequently use this idea to conduct Bayesian inference for a range of economic and financial models, employing the BPF as the base particle filtering method. In addition, Pitt *et al.* (2012), drawing on Del Moral (2004), explicitly demonstrate the unbiased property of the filtering-based likelihood estimators that are the focus of their work and, as noted earlier, investigate the role played by the number of particles in the resultant mixing of the chain. In summary, and as might be anticipated, for any given particle filter an increase in the number of particles improves the precision of the corresponding likelihood estimator (by decreasing its variance) and, hence, yields efficiency that is arbitrarily close to that associated with an MCMC algorithm that accesses the exact likelihood function. However, this accuracy is typically obtained at significant computational cost, with the recommendation of Pitt *et al.* being to choose the number of particles that minimizes the cost of obtaining a precise likelihood estimator yet still results in a sufficiently fast-mixing MCMC chain. Our aim is, in part, to extend this analysis to cater for the DPF filters derived here and to explore the performance of these new filters, both in a range of SSM settings and in comparison with a number of competing filters. However, the overarching aim is to ascertain if differences in algorithmic performance - arising from the use of different filters - translate into notable differences in the PMMH-based forecast distributions and, hence, in forecast accuracy.[2]

2.2 Forecasting with a PMMH algorithm

Given the Markovian structure of (1) and the condition provided by (2), the one-step-ahead forecast density is given by

$$p\left(y_{T+1}|y_{1:T}\right)=\int\int\int p\left(y_{T+1}|x_{T+1},\theta\right)p\left(x_{T+1}|x_{T},\theta\right)p(x_{T}|y_{1:T},\theta)p\left(\theta|y_{1:T}\right)dx_{T+1}dx_{T}d\theta. \qquad (6)$$

To produce an estimate of this density, we start with a sequence of MH draws of θ drawn using the PMMH Markov chain, which result in the discrete estimator of the posterior distribution $p\left(\theta|y_{1:T}\right)$, given by

$$\widehat{p}\left(\theta|y_{1:T}\right)=\frac{1}{MH}\sum_{i=1}^{MH}\delta_{\theta^{(i)}},$$

where $\delta_{(\cdot)}$ denotes the (Dirac) delta function, see Au and Tam (1999).[3] Then, for each draw $\theta^{(i)}$, the one-step-ahead predictive density, $p\left(y^{o}_{T+1}|y_{1:T},\theta^{(i)}\right)$, is obtained through a final iteration of the particle filter, over a grid of potential future observed values $\left\{y^{o}_{T+1}\right\}$. The estimate of (6) is found by simply averaging these one-step-ahead predictive densities, pointwise at each y^{o}_{T+1} on the grid.

3 The Filtering Algorithms

3.1 A quick review

Particle filtering involves the sequential application of importance sampling as each new observation becomes available, with the (incremental) target density at time $t+1$, being proportional to the

[2] Note, other PMCMC approaches, such as particle Gibbs, use particle *smoothing* techniques, in the production of $p\left(\theta|y_{1:T}\right)$, and bring with them additional numerical issues as a consequence.(See Lindsten, Jordan and Schön, 2014, and Chopin and Singh, 2015.) Given our focus on forecasting *per se*, as opposed to (joint) parameter and state posterior inference, we focus solely on PMMH, and the extraction from that algorithm of the filtered path of the states for use in the forecasting exercise.

[3] Strictly speaking, $\delta_{(\cdot)}$ is a generalized function, and is properly defined as a measure rather than as a function. However, we take advantage of the commonly used heuristic definition here, as is also done in, for example, Ng. *et al* (2013).

product of the measurement density, $p\left(y_{t+1}|x_{t+1},\theta\right)$, and the transition density of the state x_{t+1}, denoted by $p\left(x_{t+1}|x_t,\theta\right)$, as follows,

$$p(x_{t+1}|x_t, y_{1:t+1}, \theta) \propto p\left(y_{t+1}|x_{t+1},\theta\right) p\left(x_{t+1}|x_t,\theta\right). \tag{7}$$

Note that draws of the conditioning state value x_t are, at time $t+1$, available from the previous iteration of the filter.

Particle filters thus require the specification, at time $t+1$, of a proposal density, denoted generically here by

$$g(x_{t+1}|x_t, y_{1:t+1}, \theta), \tag{8}$$

from which the set of particles, $\{x_{t+1}^{[j]}, j = 1, ..., N\}$, are generated and, ultimately, used to estimate the filtered density as:

$$\widehat{p}(x_{t+1}|y_{1:t+1}, \theta) = \sum_{j=1}^{N} \pi_{t+1}^{[j]} \delta_{x_{t+1}^{[j]}}. \tag{9}$$

The normalized weights $\pi_{t+1}^{[j]}$ vary according to the choice of the proposal $g\left(\cdot\right)$, the approach adopted for marginalization (with respect to previous particles), and the way in which past particles are 'matched' with new particles in independent particle filter (IPF)-style algorithms (Lin *et al.*, 2005), of which the DPF is an example. In the case of the BPF the proposal density in (8) is equated to the transition density, $p(x_{t+1}|x_t,\theta)$, whilst for the APF the proposal is explicitly dependent upon both the transition density and the observation y_{t+1}, with the manner of the dependence determined by the exact form of the auxiliary filter (see Pitt and Shephard, 1999, for details). The use of both the observation and the previous state particle in the construction of a proposal distribution is referred to as 'adaptation' by the authors, with 'full' adaptation being feasible (only) when $p(y_{t+1}|x_t,\theta)$ can be computed and $p(x_{t+1}|x_t, y_{t+1}, \theta)$ is able to be sampled from directly. The UPF algorithm of van de Merwe *et al.* (2000) represents an alternative approach to adaptation, with particles proposed via an approximation to the incremental target that uses unscented transformations. For the IPF of Lin *et al.* (2005), the proposal reflects the form of $p\left(y_{t+1}|x_{t+1},\theta\right)$ in some (unspecified) way, where the term 'independence' derives from the lack of dependence of the draws of x_{t+1} on any previously obtained (and retained) draws of x_t.

With particle degeneracy (over time) being a well-known feature of filters, a resampling step is typically employed. While most algorithms, including the BPF, the UPF and the IPF, resample particles using the normalized weights $\pi_{t+1}^{[j]}$, the APF incorporates resampling directly within $g\left(\cdot\right)$ by sampling particles from a *joint* proposal, $g(x_{t+1}, k|x_t, y_{1:t+1}, \theta)$, where k is an auxiliary variable that indexes previous particles. This allows the resampling step, or the sampling of k, to take advantage of information from the newly arrived observation, y_{t+1}.

Given the product form of the target density $p(x_{t+1}|x_t, y_{1:t+1}, \theta)$ in (7), the component that is relatively more concentrated as a function of the argument x_{t+1} - either $p\left(y_{t+1}|x_{t+1},\theta\right)$ or $p\left(x_{t+1}|x_t,\theta\right)$

- will dominate in terms of determining the shape of the target density. In the case of a strong signal-to-noise ratio, meaning that the observation y_{t+1} provides significant information about the location of the unobserved state and with $p\left(y_{t+1}|x_{t+1},\theta\right)$ highly peaked around x_{t+1} as a consequence, an IPF proposal, which attempts to mimic $p\left(y_{t+1}|x_{t+1},\theta\right)$ alone, can yield an accurate estimate of $p(x_{t+1}|x_t, y_{1:t+1},\theta)$, in particular out-performing the BPF, in which no account at all is taken of y_{t+1} in producing proposals of x_{t+1}. Lin *et al.* (2005) in fact demonstrate that, in a high signal-to-noise ratio scenario, an IPF-based estimator of the mean of a filtered distribution can have a substantially smaller variance than an estimator based on either the BPF or the APF, particularly when computational time is taken into account. Since the DPF is an IPF it produces draws of x_{t+1} via the structure of the measurement density alone. The UDPF then augments the information from $p\left(y_{t+1}|x_{t+1},\theta\right)$ with information from the second component in (7). We detail these two new filters in the following section.

3.2 The new 'data-driven' filters

The key insight, first highlighted by Ng *et al.* (2013) and motivating the DPF and UDPF filters, is that the measurement y_{t+1} corresponding to the state x_{t+1} in period $t+1$ is often specified via a measurement equation,

$$y_{t+1} = h\left(x_{t+1},\eta_{t+1}\right), \tag{10}$$

for a given function $h\left(\cdot,\cdot\right)$ and *i.i.d.* random variables η_{t+1} having common pdf $p\left(\eta_{t+1}\right)$, where $h\left(\cdot,\cdot\right)$ and $p\left(\eta_{t+1}\right)$ are θ-dependent. Then, via a transformation of variables, the measurement density may be expressed as

$$p(y_{t+1}|x_{t+1},\theta) = \int_{-\infty}^{\infty} p(\eta_{t+1}) \left|\frac{\partial h}{\partial x_{t+1}}\right|^{-1}_{x_{t+1}=x_{t+1}(y_{t+1},\eta_{t+1})} \delta_{x_{t+1}(y_{t+1},\eta_{t+1})} d\eta_{t+1}, \tag{11}$$

where $x_{t+1}(y_{t+1},\eta_{t+1})$ is the unique solution to $y_{t+1} - h(x_{t+1},\eta_{t+1}) = 0$.[4] Further discussion of the properties of the representation in (11) are provided in Ng *et al.* The advantage of the representation in (11) is that properties of the delta function may be employed to manipulate the measurement density in various ways. Whereas Ng *et al.* exploit this representation within a grid-based context, where the grid is imposed over the range of possible values for the measurement error η_{t+1}, here we exploit the representation to devise new particle filtering proposals, as detailed in the following two subsections.

3.2.1 The data-driven particle filter (DPF)

With reference to (10), the DPF proposes particles by simulating replicate and independent measurement errors, $\eta_{t+1}^{[j]} \overset{i.i.d.}{\sim} p(\eta_{t+1})$, and, given y_{t+1}, transforming these draws to their implied state values $x_{t+1}^{[j]} = x_{t+1}(y_{t+1},\eta_{t+1}^{[j]})$ via solution of the measurement equation. Recognizing the role played by the

[4] Extension to a finite number of multiple roots is straightforward, and is discussed in Ng *et al.* (2013).

Jacobian in (11), the particles $x_{t+1}^{[j]}$ serve as a set of independent draws from a proposal distribution with density $g(\cdot)$ satisfying

$$g(x_{t+1}|y_{t+1},\theta)=\left|\frac{\partial h\left(x_{t+1},\eta_{t+1}\right)}{\partial x_{t+1}}\right|_{\eta_{t+1}=\eta^{*}(x_{t+1},y_{t+1})}p\left(y_{t+1}|x_{t+1},\theta\right), \tag{12}$$

where $\eta^{*}\left(x_{t+1},y_{t+1}\right)$ satisfies $y=h\left(x_{t+1},\eta^{*}\left(x_{t+1},y_{t+1}\right)\right)$. For the proposal distribution to have density $g(\cdot)$ in (12), it is sufficient to assume both partial derivatives of $h\left(\cdot,\cdot\right)$ exist and are non-zero, as is enforced in the range of applications considered here. Furthermore, and given the lack of explicit dependence of $g(\cdot)$ on x_t, the resultant sample from (12) is such that the new draw $x_{t+1}^{[j]}$ can be coupled with any previously simulated particle $x_t^{[i]}$, $i=1,2,...,N$. When the j^{th} particle $x_{t+1}^{[j]}$ is only ever matched with the j^{th} past particle $x_t^{[j]}$, for each $j=1,...,N$ and each $t=1,2,...,T$, then the (unnormalized) weight of the state draw is calculated as

$$w_{t+1}^{[j]}=\pi_t^{[j]}\frac{p\left(y_{t+1}|x_{t+1}^{[j]},\theta\right)p\left(x_{t+1}^{[j]}|x_t^{[j]},\theta\right)}{g\left(x_{t+1}^{[j]}|y_{t+1},\theta\right)}, \tag{13}$$

for $j=1,...,N$. For the DPF, therefore, we have

$$w_{t+1}^{[j]}=\pi_t^{[j]}\left|\frac{\partial h}{\partial x_{t+1}}\right|^{-1}_{\eta_{t+1}=\eta_{t+1}^{[j]},\ x_{t+1}=x_{t+1}^{[j]}}p\left(x_{t+1}^{[j]}|x_t^{[j]},\theta\right), \tag{14}$$

for $j=1,2,...,N$, where $x_0^{[j]}\overset{iid}{\sim}p\left(x_0|\theta\right)$, $\pi_0^{[j]}=\frac{1}{N}$, and the filtering weights $\pi_{t+1}^{[j]}$, in (9), are produced sequentially as

$$\pi_{t+1}^{[j]}\propto w_{t+1}^{[j]} \tag{15}$$

for all $j=1,2,...,N$, with $\sum_{j=1}^{N}\pi_t^{[j]}=1$ for each t. In addition, and as in any particle filtering setting (see, for example, Doucet and Johansen, 2011), the iteration then provides component $t+1$ of the estimated likelihood function as

$$\widehat{p}_u(y_{t+1}|y_{1:t},\theta)=\sum_{j=1}^{N}w_{t+1}^{[j]}, \tag{16}$$

with each $w_{t+1}^{[j]}$ as given in (13).

Alternatively, as highlighted by Lin *et al.* (2005), the j^{th} particle at $t+1$, could be matched with *multiple* previous particles from time t. In this case, define $w_{t+1}^{[j][i]}$ as the (unnormalized) weight corresponding to a match between $x_t^{[i]}$ and $x_{t+1}^{[j]}$,

$$w_{t+1}^{[j][i]}=\pi_t^{[i]}\frac{p\left(y_{t+1}|x_{t+1}^{[j]},\theta\right)p\left(x_{t+1}^{[j]}|x_t^{[i]},\theta\right)}{g\left(x_{t+1}^{[j]}|y_{1:t},\theta\right)},$$

for any $i=1,2,...,N$ and $j=1,2,...,N$. Next, denote L *distinct cyclic* permutations of the elements in the sequence $(1,2,...,N)$ by $K_l=(k_{l,1},...,k_{l,N})$, for $l=1,...,L$. For each permutation l, the j^{th} particle $x_{t+1}^{[j]}$ is matched with the relevant past particle indicated by $x_t^{[k_{l,j}]}$. Then, the final (unnormalized) weight associated with $x_{t+1}^{[j]}$ is the simple average, $w_{t+1}^{[j]} = \frac{1}{L}\sum_{l=1}^{L} w_{t+1}^{[j][k_{l,j}]}$. Thus, in the DPF with multiple matching case, for $j = 1, 2, ..., N$, we have

$$w_{t+1}^{[j]} = \frac{1}{L}\left|\frac{\partial h}{\partial x_{t+1}}\right|^{-1}_{\eta_{t+1}=\eta_{t+1}^{[j]},\ x_{t+1}=x_{t+1}^{[j]}} \sum_{l=1}^{L}\left[\pi_t^{[k_{l,j}]} p\left(x_{t+1}^{[j]}|x_t^{[k_{l,j}]}, \theta\right)\right], \tag{17}$$

with $\pi_t^{[j]}$ available from the previous iteration of the filter. Accordingly, as for the $L = 1$ matching case in (15), the $\pi_{t+1}^{[j]}$ are then set proportional to the $w_{t+1}^{[j]}$ in (17), with $\sum_{j=1}^{N}\pi_{t+1}^{[j]} = 1$. We consider this suggestion in Section 4, and document the impact of the value of L on filter performance.[5] To ensure ease of implementation, pseudo code for the DPF algorithm is provided as Algorithm 1 in Appendix A.1. Note that, as we implement the resampling of particles at each iteration of all filters employed in Sections 4 and 5, these steps are included as steps 8 and 9 in Algorithm 1.

The DPF, when applicable, thus provides a straightforward and essentially automated way to estimate the likelihood function, in which only the measurement equation is used in the generation of new particles.[6] However, its performance depends entirely on the extent to which the current observation is informative in identifying the unobserved state location. This motivates the development of the UDPF, in which proposed draws are informed by both the current observation and a previous state particle.

3.2.2 The unscented data-driven particle filter (UDPF)

The UDPF uses a Gaussian approximation to the measurement density, $p(y_{t+1}|x_{t+1}, \theta)$,

$$\widehat{p}(y_{t+1}|x_{t+1}, \theta) \propto \frac{1}{\widehat{\sigma}_{M,t+1}}\phi\left(\frac{x_{t+1} - \widehat{\mu}_{M,t+1}}{\widehat{\sigma}_{M,t+1}}\right). \tag{18}$$

The terms $\widehat{\mu}_{M,t+1}$ and $\widehat{\sigma}^2_{M,t+1}$ denote the (approximated) first and second (centred) moments (of x_{t+1}) implied by an unscented transformation of η_{t+1} to x_{t+1}, for a given value of y_{t+1}, and where the subscript M is used to reference the measurement equation via which these moments are produced. Our motivation for using the unscented method, including all details of the computation of the moments in this case, is provided in Appendix A.2. The method derives from combining an assumed Gaussian transition density, given by

$$p(x_{t+1}|x_t, \theta) = \frac{1}{\widehat{\sigma}_{P,t+1}}\phi\left(\frac{x_{t+1} - \widehat{\mu}_{P,t+1}^{[j]}}{\widehat{\sigma}_{P,t+1}^{[j]}}\right), \tag{19}$$

where $\widehat{\mu}_{P,t+1}^{[j]}$ and $\widehat{\sigma}_{P,t+1}^{2[j]}$ are assumed known, given particle $x_t^{[j]}$, and the subscript P references the predictive transition equation to which these moments apply. If needed, a Gaussian approximation

[5] We note that the choice of $L = N$ yields the incremental weight corresponding to the so-called marginal version of the filter. See Klaas, de Freitas and Doucet (2012).

[6] This idea of generating particles using only information from the observation and the measurement equation is, in fact, ostensibly similar to notions of fiducial probability (see e.g. Hannig, Iyer, Lai and Lee, 2016). However, in this case, whilst the proposal density in (12) for the latent state x_{t+1} is obtained without any knowledge of the predictive density (or 'prior') given by $p(x_{t+1}|y_{1:t}, \theta)$, the resampling of the proposed particles according to either (14) or (17), means that the resampled draws themselves *do* take account of this predictive information, and thus appropriately represent the filtered distribution as in (9).

(e.g. via a further unscented transformation, in which case both $\widehat{\mu}_{P,t+1}^{[j]}$ and $\widehat{\sigma}_{P,t+1}^{2[j]}$ may also depend upon $x_t^{[j]}$) of a non-Gaussian state equation may be accommodated. Having obtained the Gaussian approximation in (18) and applying the usual conjugacy algebra, a proposal density is constructed as

$$g(x_{t+1}|x_t^{[j]}, y_{t+1}, \theta) = \frac{1}{\widehat{\sigma}_{t+1}}\phi\left(\frac{x_{t+1} - \widehat{\mu}_{t+1}^{[j]}}{\widehat{\sigma}_{t+1}^{[j]}}\right), \tag{20}$$

where $\widehat{\mu}_{t+1}^{[j]} = \left\{\widehat{\sigma}_{P,t+1}^{2[j]}\widehat{\mu}_{M,t+1} + \widehat{\sigma}_{M,t+1}^{2}\widehat{\mu}_{P,t+1}^{[j]}\right\} / \left\{\widehat{\sigma}_{P,t+1}^{2[j]} + \widehat{\sigma}_{M,t+1}^{2}\right\}$ and $\widehat{\sigma}_{t+1}^{2[j]} = \widehat{\sigma}_{M,t+1}^{2}\widehat{\sigma}_{P,t+1}^{2[j]}/\left\{\widehat{\sigma}_{P,t+1}^{2[j]} + \widehat{\sigma}_{M,t+1}^{2}\right\}$ are the requisite moments of the Gaussian proposal, with the bracketed superscript $[j]$ used to reflect the dependence on the j^{th} particle $x_t^{[j]}$. A resulting particle draw $x_{t+1}^{[j]}$ from the proposal in (20) is then weighted, as usual, relative to the target, with the particle weight formula given by,

$$w_{t+1}^{[j]} = \pi_t^{[j]}\frac{p\left(y_{t+1}|x_{t+1}^{[j]}, \theta\right)p\left(x_{t+1}^{[j]}|x_t^{[j]}, \theta\right)}{\frac{1}{\widehat{\sigma}_{t+1}^{[j]}}\phi\left(\frac{x_{t+1}^{[j]} - \widehat{\mu}_{t+1}^{[j]}}{\widehat{\sigma}_{t+1}^{[j]}}\right)}. \tag{21}$$

The corresponding UDPF likelihood estimator is calculated as per (16). Pseudo code for the UDPF is provided in Algorithm 2 in Appendix A.1, and proof of the unbiasedness of both data-driven filters is given in Appendix A.3.

4 Simulation Experiments

In this section, two different simulation exercises are undertaken. In the first experiment we investigate, in a controlled setting, the performance of the five filters, each described in Section 3, in terms of their ability to produce the posterior distribution for the model parameters. In particular, in Section 4.4, we document both the computational time and mixing performance of the relevant PMMH algorithm, for three different state space models: the linear Gaussian (LG) model that is foundational to all state space analysis, and two non-linear, non-Gaussian models referenced in the Introduction that feature in the empirical finance literature, namely: the stochastic conditional duration (SCD) specification (used to model the duration between stock market trades; Bauwens and Veredas, 2004) and the stochastic volatility (SV) specification (used to model the volatility in financial returns; Shephard, 2005). Notably, different signal-to-noise ratios are entertained for all three models. In Section 4.5, the forecast accuracy of the estimate of (6) yielded by each filtering method is considered under, respectively, correct and incorrect specification of the true DGP. Critically, having documented the differential performance of the alternative filters in Section 4.4, we then deliberately hold both the number of particles and the number of MCMC draws fixed in Section 4.5, in order to gauge the impact, or otherwise, of this differential performance on forecast accuracy.

4.1 Simulation design and PMMH evaluation methods

Before detailing the specific design scenarios adopted for the simulation exercises, we first define the signal-to-noise ratio (SNR) as

$$SNR = \sigma_x^2/\sigma_m^2, \tag{22}$$

where σ_x^2 is the unconditional variance of the state variable. In the LG setting σ_m^2 corresponds directly to variance of the additive measurement noise. In the two non-linear models, a transformation of the measurement equation is employed to enable the calculation of σ_m^2, now given by the variance of the (transformed) measurement error that results, to be obtained either analytically (for the SV model) or using deterministic integration (for the SCD model). The details of the relevant transformations are provided in Sections 4.2.2 and 4.2.3, respectively. The quantity in (22) measures the strength of signal relative to the background noise in the (appropriately transformed) SSM, for given fixed parameter values.[7]

A design scenario is defined by the combination of the model and corresponding model parameter settings that achieve a given value for (22), either low or high. What constitutes a particular level for SNR is model-specific, with values chosen (and reported below) that span the range of possible SNR values that still accord with empirically plausible data. If a particular design has a high SNR, this implies that observations are informative about the location of the unobserved state. The DPF is expected to perform well in this case, in terms of precisely estimating the (true) likelihood value, with the impact of the use of multiple matching being of particular interest. Conversely, as the BPF proposes particles from the state predictive distribution, it is expected to have superior performance to the DPF when the SNR is low. Exploiting both types of information at the same time, the UDPF, APF and UPF methods are anticipated to be more robust to the SNR value. However, when assessing PMMH performance and, ultimately forecast performance, there is a more complex relationship between the filter performance and the SNR of the DGP, given that estimation of the likelihood function takes place across the full support of the unknown parameters.

The PMMH assessment draws on the insights of Pitt *et al.* (2012). If the likelihood is estimated precisely, the mixing of the Markov chain will be as rapid as if the true likelihood were being used, and the estimate of any posterior quantity will also be accurate as a consequence. However, increasing the precision of the likelihood estimator by increasing the number of particles used in the filter comes at a computational cost. Equivalently, if a poor, but computationally cheap, likelihood estimator is used within the PMMH algorithm, this will typically slow the mixing of the chain, meaning that for a given number of Markov chain iterates, the estimate of any posterior quantity will be less accurate. Pitt *et al.* suggest choosing the particle number that minimizes the so-called computing time: a measure that takes into account both the cost of obtaining the likelihood estimator and the speed of

[7] A comparable quantity that is applicable to the non-linear case is defined by $SNR^* = \sigma_x^2/V$, where V is the curvature of $\log(p(y_t|x_t,\theta))$, see Giordani *et al.* (2011).

mixing of the MCMC chain. They show that the 'optimal' number of particles is that which yields a variance for the likelihood estimator of 0.85, at the true parameter vector.[8]

For each design scenario (i.e. model and SNR level), and for each particular filter, the PMMH algorithm is used to produce a Markov chain with $MH = 110,000$ iterations, with the first $10,000$ iterations being discarded as burn-in. The MCMC draws are generated from a random walk proposal, with the covariance structure of the proposal adapted using Algorithm 1 of Müller (2010). Determining the optimal number of particles, N_{opt}, for any particular design scenario and for any specific filter, involves producing R_0 independent replications of the likelihood estimate $\widehat{p}_u^{(r)}(y_{1:T}|\theta_0)$, each evaluated at the true parameter (vector) θ_0 and based on a selected number of particles, N_s. Then, the optimum number of particles, denoted by N_{opt}, is chosen according to

$$N_{opt} = N_s \times \frac{\widehat{\sigma}^2_{N_s}}{0.85}, \tag{23}$$

where $\widehat{\sigma}^2_{N_s}$ denotes the variance of the R_0 likelihood estimates. In other words, the (somewhat arbitrarily selected) initial number of particles, N_s, is scaled according to the extent to which the precision that it is expected to yield (as estimated by $\widehat{\sigma}^2_{N_s}$) varies from the value of 0.85 that is sought. We track the time taken to compute the likelihood estimate at each of the MH iterations using the given filter with N_{opt} particles. We then record the average likelihood computing time (ALCT) over these iterations, as well as the inefficiency factor (IF) for each parameter. In the usual way, the IF for a given parameter can be interpreted as the sampling variance of the mean of the correlated MCMC draws of that parameter relative to the sampling variance of the mean of a hypothetical set of independent draws. Values greater than unity thus measure the loss of precision (or efficiency) incurred due to the dependence in the chain.[9]

4.2 Models, SNR ratios and priors

In this section, we outline the three models used in the simulation experiments. The parameter values and associated values for SNR are contained in Table 1.

4.2.1 The linear Gaussian (LG) model

The LG model is given by

$$y_t = x_t + \sigma_\eta \eta_t \tag{24}$$

$$x_t = \rho x_{t-1} + \sigma_v v_t, \tag{25}$$

with η_t and v_t mutually independent *i.i.d.* standard normal random variables. Data is generated using $\rho = 0.4$ and $\sigma_v = 0.92$. The value of σ_η is set to achieve two values for SNR (low and high), as

[8] Note, while the optimal number of particles may be computed within a simulation context, as in the current section, implementation in an empirical setting requires a preliminary estimate of the parameter (vector) at which this computation occurs.

[9] We highlight once again that the aim of this paper is not to optimize the performance of any PMMH algorithm for its own sake. Hence, we are not exploiting any of the more recent contributions to the literature (including those cited in the Introduction) in which performance improvements have been achieved via various means.

Table 1: Parameters values used in the simulation exercises for the LG, SCD and SV models. The corresponding signal-to-noise ratio (SNR) for each scenario is shown in the bottom row.

	PANEL A: LG			PANEL B: SCD			PANEL C: SV	
	Low	High		Low	High		Low	High
σ_η	2.24	0.45	α	0.67	6.67	ϕ	-6.61	-4.24
ρ	0.40	0.40	β	1.50	0.15	ρ	0.20	0.60
σ_v	0.92	0.92	ϕ	-1.10	-1.10	σ_v	0.70	1.40
			ρ	0.74	0.74			
			σ_v	0.65	0.65			
SNR	0.20	5.00		0.50	10.00		0.10	0.60

recorded in Panel A of Table 1. In the PMMH exercises detailed in Section 4.4, where the parameters are treated as unknown, the parameter $\theta = (\log(\sigma_\eta^2), \rho, \log(\sigma_v^2))'$ is sampled (thereby restricting the simulated draws of σ_v^2 and σ_η^2 in the resulting Markov chains to be positive), with a normal prior distribution assumed as $\theta \sim N(\mu_0, \Sigma_0)$ with $\mu_0 = (\log(0.7), 0.5, \log(0.475))'$ and $\Sigma_0 = I_n$. The same prior is used in both high and low SNR settings, and is in the spirit of the prior used in Flury and Shephard (2011).

4.2.2 The stochastic conditional duration (SCD) model

The SCD model is given by

$$y_t = \exp(x_t)\eta_t \tag{26}$$

$$x_t = \phi + \rho x_{t-1} + \sigma_v v_t, \tag{27}$$

with $v_t \sim i.i.d.N(0,1)$ independent of η_t, and with η_t being *i.i.d.* from a gamma distribution with shape parameter α and rate parameter β. Taking the logarithms of both sides of (26) yields a transformed measurement equation that is linear in the state variable x_t, i.e. $\log(y_t) = x_t + \varepsilon_t$, where $\varepsilon_t = \log(\eta_t)$. The value of $\sigma_m^2 = var(\varepsilon_t)$ required to report the SNR in Table 1 is obtained numerically. The initial state is taken as the long run distribution of the state implied by choosing $|\rho| < 1$, that is $x_0 \sim N\left(\frac{\phi}{1-\rho}, \frac{\sigma_v^2}{(1-\rho)^2}\right)$. For the PMMH exercise detailed in Section 4.4, the parameter vector $\theta = (\log(\alpha), \log(\beta), \phi, \rho, \log(\sigma_v^2))$ is used to ensure the positivity of draws for each α, β and σ_v^2. As with the LG setting, a normal prior is adopted with $\theta \sim N(\mu_0, \Sigma_0)$, but now with $\mu_0 = (-0.8, 0.5, \log(0.5), \log(2), \log(1))'$ and $\Sigma_0 = I_n$. This prior is again held constant over the two SNR settings (low and high) used to assess PMMH performance.

4.2.3 The stochastic volatility (SV) model

The SV model is given by

$$y_t = \exp(x_t/2)\eta_t \tag{28}$$

$$x_t = \phi + \rho x_{t-1} + \sigma_v v_t, \tag{29}$$

with η_t and v_t once again mutually independent sequences of *i.i.d.* standard normal random variables. To fix the SNR, the measurement equation is transformed to $\log(y_t^2) = x_t + \varepsilon_t$, where $\varepsilon_t = \log(\eta_t^2)$. In this case, $var(\varepsilon_t) = 4.93$, corresponding to the quantity σ_m^2 in (22). The initial state distribution is specified from the stationary distribution, with $x_0 \sim N(\frac{\phi}{1-\rho}, \frac{\sigma_v^2}{(1-\rho)^2})$. For the PMMH exercise, a normal prior is adopted for $\theta = (\phi, \rho, \log(\sigma_v^2))'$, with $\theta \sim N(\mu_0, \Sigma_0)$, where $\mu_0 = (-4.6, 0.8, \log(0.5))'$ and $\Sigma_0 = I_n$. This prior is used under both SNR settings.

4.3 Filter implementation details

The DPF and the UPDF are explained in detail in Sections 3.2.1 and 3.2.2, respectively. The DPF is implemented with both $L = 1$ and $L = 30$ matches. Implementation of the BPF is standard, with details available from many sources (e.g. Gordon *et al.*, 1993, and Creal, 2012). The APF, on the other hand, may be implemented in a variety of different ways, depending upon the model structure and the preference of the analyst. For the models considered in this paper, so-called full adaptation is feasible (only) for the LG model, and hence we report results for this version of the filter (referred to as FAPF hereafter) in that case. For all three models, we also experimented with an alternative version of APF (in which full adaptation is not exploited) where the proposal distribution is given by $g(x_{t+1}, k|x_t, y_{1:t+1}, \theta) = p(y_{t+1}|\mu(x_t^{[k]}, \theta))p(x_{t+1}|x_t^{[k]}, \theta)$, where $\mu(x_t^{[k]})$ is the conditional mean $E(x_{t+1}|x_t^{[k]}, \theta)$, and k is a discrete auxiliary variable (see Pitt and Shephard, 1999, for details). For the SV model, we explored a third version of APF based on a second order Taylor's series expansion of $\log(p(y_{t+1}|x_{t+1}, \theta))$ around the maximum of the measurement density. This approach yields an approximation of the likelihood component, denoted by $g(y_{t+1}|x_{t+1}, \theta)$, which is then used to form a proposal distribution, $g(x_{t+1}, k|x_t, y_{1:t+1}, \theta) = g(y_{t+1}|x_{t+1}, \mu(x_t^{[k]}, \theta))p(x_{t+1}|x_t^{[k]}, \theta)$. (For more details, see Pitt and Shephard, 1999, and Smith and Santos, 2006.) For the particular non-linear models explored here, however, both non-fully-adapted APF methods resulted in very unstable likelihood estimates. Hence, these filters were not pursued further in either the documentation of PMMH performance results or the production of PMMH-based forecast distributions.[10]

As is standard knowledge, the KF is a set of recursive equations suitable for the LG model that enable calculation of the first two moments of the distribution of the unobserved state variables given progressively observed measurements. In a non-linear setting, the unscented Kalman filter uses approximate Gaussian distributions obtained from the unscented transformations applied within

[10] Further details on these results can be obtained from the authors on request.

the recursive KF structure, to approximate each of the (non-Gaussian) filtered state distributions. In contrast, the UPF that is implemented in our setting, uses approximate Gaussian distributions for the proposal distributions in (8) with moments produced by the unscented transformations, and with the conditioning on each new observation y_{t+1} obtained as if the model were an LG model with moments that match those of the conditional distributions defined by $p\left(y_{t+1}|x_{t+1},\theta\right)$ and $p\left(x_{t+1}|x_{t},\theta\right)$. Further discussion of the UPF is provided in van de Merwe *et al.* (2000).

4.4 PMMH performance: Simulation results

At each MCMC iteration, the particle filter based on N_{opt} is used to estimate the likelihood function conditional on the set of parameter values drawn at that iteration. The value of N_{opt}, however, is determined (via the preliminary exercise described in Section 4.1, with $R_0 = 100$ replications and $N_s = 1000$ particles) at the true parameter values only and, hence, is influenced by the SNR associated with the true data generating process. Thus, when considering the performance of the filters within an MCMC algorithm two things are required: 1) efficient performance at the SNR for the true data, leading to a small value of N_{opt}; plus 2) some robustness in performance to the SNR, since the movement across the parameter space (within the chain) effectively changes the SNR under which the likelihood function is computed at each point. A small value of N_{opt} will, *ceteris paribus*, tend to produce a small value for the ALCT and, thus, ease the computational burden. However, a lack of robustness of the filter will lead to inaccurate likelihood estimates and, hence poor mixing in the chain. Both the ALCT and the IF thus need to be reported for each filter, and for each model, with the preferable filter being that which yields acceptable mixing performance in a reasonable time across for all three models. The results documented in this section are based on a sample size of $T = 250$, reflecting the need for at least a moderate sample size when comparing the performance of competing *inferential* algorithms in a state space setting.

The PMMH results for the LG model are presented in Table 2. As is consistent with expectations, under the high SNR setting, the optimum number of particles for the BPF is much larger than that for the DPF. This then translates into higher values for ALCT for the BPF than for the DPF, when a single match only ($L = 1$) is used. Further reduction in N_{opt} is yielded via the multiple matching ($L = 30$), via the extra precision that is produced from the averaging process. However, this comes at a distinct cost in computational time, with the gain of the DPF over the BPF, in terms of ALCT, lost as a consequence. In the low SNR setting, also as anticipated, the basic DPF (for either value of L) does not produce gains over the BPF, either in terms of N_{opt} or ALCT.

In contrast to the variation in the performance of the DPF - relative to the BPF - over the SNR settings, the UDPF is uniformly superior to the BPF in terms of N_{opt}, with the increase in computational cost associated with the likelihood estimation (as a consequence of having to perform the unscented transformations) resulting in only a slightly larger value for ALCT (relative to that

Table 2: LG model: The optimal number of particles, average likelihood computing time (ALCT) and the inefficiency factor (IF) are reported for the PMCMC algorithm using DPF (with $L = 1$ and 30 matches), BPF, UPF, UDPF and FAPF to produce the likelihood estimator. Data is simulated from the model in (22) and (23) with SNR = 0.2 in the top panel and SNR = 5 in the bottom panel.

PMMH results under a low SNR setting

	N_{opt}	ALCT	IF		
			σ_η^2	ρ	σ_v^2
DPF ($L = 1$)	379	0.100	241.4	325.2	248.6
DPF ($L = 30$)	348	0.647	300.2	321.3	313.1
BPF	18	0.017	184.5	67.7	178.6
UPF	57	0.057	128.2	88.3	144.5
UDPF	4	0.065	134.2	63.3	149.9
FAPF	2	0.022	163.8	98.4	181.6

PMMH results under a high SNR setting

	N_{opt}	ALCT	IF		
			σ_η^2	ρ	σ_v^2
DPF ($L = 1$)	168	0.084	33.3	31.9	33.3
DPF ($L = 30$)	143	0.323	33.1	33.6	35.9
BPF	2750	0.254	20.9	20.0	20.4
UPF	70	0.066	31.6	31.6	32.2
UDPF	23	0.060	37.4	35.5	39.9
FAPF	11	0.035	35.0	35.4	39.8

Table 3: SCD model: The optimal number of particles, average likelihood computing time (ALCT) and the inefficiency factor (IF) are reported for the PMCMC algorithm using DPF (with $L = 1$ and 30 matches), BPF, UPF and UDPF to produce the likelihood estimator. Data is simulated from the model in (24) and (25) with SNR = 0.5 in the top panel and SNR = 10 in the bottom panel.

PMMH results under a low SNR setting

	N_{opt}	ALCT	IF				
			ϕ	ρ	σ_v^2	α	β
DPF ($L = 1$)	1469	0.346	50.3	61.4	34.7	39.3	52.8
DPF ($L = 30$)	1296	2.100	47.7	45.6	38.8	37.2	45.9
BPF	184	0.064	58.3	57.3	46.1	40.0	42.6
UPF	398	0.239	55.2	67.1	49.6	42.3	66.3
UDPF	119	0.065	45.9	55.9	52.3	36.8	55.1

PMMH results under a high SNR setting

	N_{opt}	ALCT	IF				
			ϕ	ρ	σ_v^2	α	β
DPF ($L = 1$)	177	0.060	55.6	51.7	48.0	49.2	44.4
DPF ($L = 30$)	159	0.381	40.8	43.9	47.2	48.9	36.0
BPF	1011	0.218	45.8	44.6	46.1	62.0	34.5
UDPF	73	0.068	62.4	67.6	69.5	72.6	46.8
UPF	118	0.113	65.1	58.7	54.7	54.8	54.6

Table 4: SV model: The optimal number of particles, average likelihood computing time (ALCT) and the inefficiency factor (IF) are reported for the PMCMC algorithm using DPF (with $L = 1$ and 30 matches), BPF, UPF and UDPF to produce the likelihood estimator. Data is simulated from the model in (26) and (27) with SNR = 0.1 in the top panel and SNR = 0.6 in the bottom panel.

PMMH results under a low SNR setting

	N_{opt}	ALCT	IF		
			ϕ	ρ	σ_v^2
DPF (L=1)	1767	0.534	22.5	22.4	19.1
DPF (L=30)	1548	3.370	21.4	21.4	18.6
BPF	275	0.067	14.8	14.8	16.2
UPF	244	0.133	22.4	22.3	21.4
UDPF	245	0.085	14.6	14.6	14.0

PMMH results under a high SNR setting

	N_{opt}	ALCT	IF		
			ϕ	ρ	σ_v^2
DPF ($L = 1$)	1013	0.265	14.9	15.4	15.5
DPF ($L = 30$)	915	1.889	17.6	18.3	16.4
BPF	605	0.104	16.9	16.6	14.9
UPF	686	0.256	16.4	16.5	18.8
UDPF	568	0.156	12.6	12.9	13.8

for the BPF) in the low SNR case. Moreover, the UDPF yields very similar values of N_{opt} to the analytically available FAPF and values for ALCT that are not much higher. The values of N_{opt} for the UDPF are also much lower than those for the UPF, with ALCT being only slightly larger for the former in the low SNR case.

As one would anticipate, given that N_{opt} for each filter is deliberately selected to ensure a given level of accuracy in the estimation of the likelihood (albeit at the true parameter values only), the variation in the IFs (for any given parameter) across the different filters is not particularly marked. That said, there are still some differences, with the UDPF, along with the UPF, being the best performing filters overall, when both SNR scenarios in this LG setting are considered, and the DPF (for both values of L) being the most inefficient filter in the low SNR case.

The PMMH results for the SCD and SV models are presented in Table 3 and 4 respectively. Both sets of results are broadly similar to those for the LG model in terms of the relative performance of the methods, remembering that the FAPF is not applicable in the non-linear case and all other versions of the APF are eschewed due to the poor likelihood estimation results cited earlier. For the SCD model, the conclusions drawn above regarding the relative performance of the BPF and DPF filters apply here also. In this case, however, when all three factors: robustness to SNR, ALCT value and IF value are taken into account, the UDPF is uniformly superior to all other filters. For the SV model, as the 'high' SNR value appears relatively small, set as such to ensure that the model produces empirically plausible data, the DPF has less of a comparative advantage over the BPF. However, the UDPF is competitive with the (best performing) BPF in both settings, according to ALCT, and is uniformly superior to all other filters according to the IF values.

Overall then, when robustness to SNR, computation time and chain performance are all taken into account the UDPF is the preferred choice for the experimental designs considered here. We now address the question of what difference, if any, this superiority in (algorithmic) performance makes at the forecasting level.

4.5 Forecast performance: Simulation results

The impact of the particle filter on forecasting is explored in the context of estimating the SV model described previously in Section 4.2.3. We *simulate* the data, however, under two different scenarios - one where the SV model is correctly specified, and the other where the SV model does not correspond to the true DGP. In the first case, data are simulated under the SV model in (28) and (29), using the low SNR setting shown in Panel C of Table 1. In this second case, the data follows (a discrete approximation to) a bivariate jump diffusion process, with independent random jumps sporadically occurring, in the price and/or volatility process (see Duffie, Pan and Singleton, 2000). In this context, a price jump relates to large observed deviation from the expected return, whereas a volatility jump corresponds to an unusually large deviation in the underlying volatility process. We refer to this DGP as the stochastic volatility with independent jumps (SVIJ) model.

According to the SVIJ model the observed value y_t is generated as

$$y_t = \sqrt{x_t}\zeta_t^p + Z_t^p \Delta N_t^p \tag{30}$$

$$x_t = \kappa\theta + (1-\kappa)x_{t-1} + \sigma_v\sqrt{x_t}\zeta_t^x + Z_t^x \Delta N_t^x, \tag{31}$$

with ζ_t^p and ζ_t^x being independent sequences of *i.i.d.* standard normal random variables. The jump components of the measurement equation and the state equation are both composed from two separate parts: jump occurrence and size. The jump occurrence sequences have elements denoted by ΔN_t^p and ΔN_t^x, respectively, and are each independent *i.i.d.* Bernoulli random variables taking the value one with 15% and 20% probability, respectively. The size of each jump in the state equation, Z_t^x, is generated from an exponential distribution, with a mean value of 0.02. The size of a price jump, Z_t^p, is generated as $Z_t^p = S_t \exp(M_t)$, where S_t is equal to either $+1$ or -1 with equal probability, and where M_t is *i.i.d.* from a normal distribution with mean zero and variance equal to 0.5. These numerical values were selected to accord with estimated values obtained from the empirical study of the S&P 500 index undertaken in Maneesoonthorn, Forbes and Martin (2017).

In both the correctly and incorrectly specified cases, a single time series of length $T = 750$ is generated, with the first 500 observations used to produce competing PMMH-based estimates of the posterior distribution $p(\theta|y_{1:500})$, where $\theta = (\phi, \rho, \sigma_v^2)'$, i.e. the parameters of the estimated SV model. At each PMMH iteration, a candidate draw of the transformed parameter, $\widetilde{\theta} = (\phi, \rho, \log(\sigma_v^2))'$,

Table 5: The inefficiency factors obtained from the PMMH algorithms, with the DPF, BPF, UPF and UDPF used to produce the likelihood estimator. The data is simulated from the SV model (columns 2-4) and the SVIJ model (columns 5-7).

	IF					
	SV			SVIJ		
	ϕ	ρ	σ_v^2	ϕ	ρ	σ_v^2
DPF	386.5	356.6	370.1	354.0	353.6	301.9
BPF	23.6	23.6	20.6	20.8	20.8	18.7
UPF	80.2	80.5	72.5	40.7	10.8	32.9
UDPF	23.0	23.0	22.1	18.8	18.8	16.1

is generated from a random walk proposal. The prior distribution of $\widetilde{\theta}$ is specified as: $\phi \sim N(0, 10)$, $\rho \sim Beta(20, 1.5)$, $\log(\sigma_v^2) \sim N(0, 10)$. For each of the four remaining alternative filters, DPF (with $L = 1$), UDPF, BPF and UPF, $N = 300$ particles are used to estimate the likelihood function at each PMMH iteration, and $MH = 5000$ iterations drawn. By holding the number of particles used in each filter fixed at a common value, the resulting efficiency associated with the estimated likelihood function, and hence the PMMH algorithm itself (for a given number of MCMC draws), will be different. The impact of controlling the number of particles on the IFs of the resultant Markov chains is evident in Table 5, where it is clear that some Markov chains are more efficient than others. Notably, the DPF is relatively inefficient compared to the other three filters and, overall, the UDPF continues to exhibit the superior performance documented in the previous section.

We then proceed to estimate the competing (marginal) one-step-ahead forecast distributions, for each of 250 subsequent periods, each time following the procedure described in Section 2.2.[11] Due to the singularity of the square root at zero, we produce forecast distributions for the transformed measurement $\log\left(y_{T+k}^2\right)$. Having produced these competing forecast distributions, their performance is measured using the average log score, which we denote by ALS and calculate as the average of the logarithm of each estimated predictive density $\widehat{p}\left(\log\left(y_{T+k}^2\right) | y_{1:T+k-1}\right)$, evaluated at the subsequently 'realised' value, $\log\left(\left(y_{T+k}^{obs}\right)^2\right)$, for $k = 1, 2, ..., 250$. We also compute the average absolute difference between the log score produced under the BPF and that produced by each of the other three filters. We denote this average absolute difference in log scores by ADLS.

Despite the four filters having quite different inefficiency factors, we find that the forecast accuracy is virtually unaffected by which filter is used, and irrespective of whether the fitted model is correctly or incorrectly specified. Figures 1 and 2 illustrate the estimated forecast distributions of the first out-of-sample period (i.e. with $T = 500$ and $k = 1$), when the data are generated by the SV DGP and the SVIJ DGP, respectively. The top panels in Figures 1 and 2 display all of the estimated conditional one-step-ahead forecast distributions associated with each of the filters, i.e. all of the

[11] Due to the intensive nature of the PMMH algorithm, the posterior draws of θ are refreshed only after 50 forecast periods.

Table 6: ALS and ADLS, obtained from estimating the SV model using PMMH with four different filters. The simulated data is simulated from the the SV (columns 2 and 3) and SVIJ (columns 4 and 5) specification, respectively.

Log score summaries

DGP	SV		SVIJ	
	ALS	ADLS	ALS	ADLS
DPF	-2.2053	0.0240	-2.1862	0.0298
BPF	-2.2071	0.0000	-2.1854	0.0000
UPF	-2.2070	0.0047	-2.1852	0.0035
UDPF	-2.2072	0.0032	-2.1856	0.0035

$\widehat{p}\left(\log\left(y^2_{T+k}\right)|y_{1:T+k-1}, \theta^{(i)}\right)$ for $i = 1, 2, ..., 5000$ MH iterations and for each filter. While all of the conditional forecast distributions produced by each of the four filters appear to be centered around a similar location, the DPF and UPF produce more varied conditional forecasts than do either the BPF or UDPF. However, since the marginal one-step-ahead forecast distribution is produced by integrating out the uncertainty associated with the unknown parameters, much of the variation between the conditional forecast distributions is eliminated through the averaging procedure. Hence, the competing estimated marginal one-step-ahead forecast distributions, shown in the bottom panels of Figures 1 and 2 for the correctly and incorrectly specified SV models, respectively, are visually indistinguishable from each other. In addition, as shown in Table 6, we find no difference (to two decimal places) in the ALS produced from the 250 one-step-ahead forecasts, with similarly negligible differences obtained for the ADLS.

5 Empirical Illustration

In this section we consider the production of forecast distributions of log-squared returns from an SV model for daily continuously compounded S&P500 returns, based on data from April 6, 2016 to April 2, 2019. The time series plot of the 754 observations from the sample period, shown in Figure 3, suggests that a reasonably sophisticated model such as that in (30) and (31) may be appropriate. However, given the robustness of the forecasts to model misspecification (as documented above), we use the simpler SV model in (28) and (29) to produce the forecasts. The prior distribution is the same as that detailed in Section 4.2.3, and the forecasting performance of each of the filters is produced using the PMMH procedure described in Section 4.5, with each filter implemented using $N = 300$ particles.

The first $T = 500$ observations are used to produce an initial one-step-ahead marginal predictive distribution, corresponding to day $T + 1 = 501$ (April 2, 2018). This process is then repeated for each subsequent period, using an expanding in-sample window and resulting in a total of 254 one-

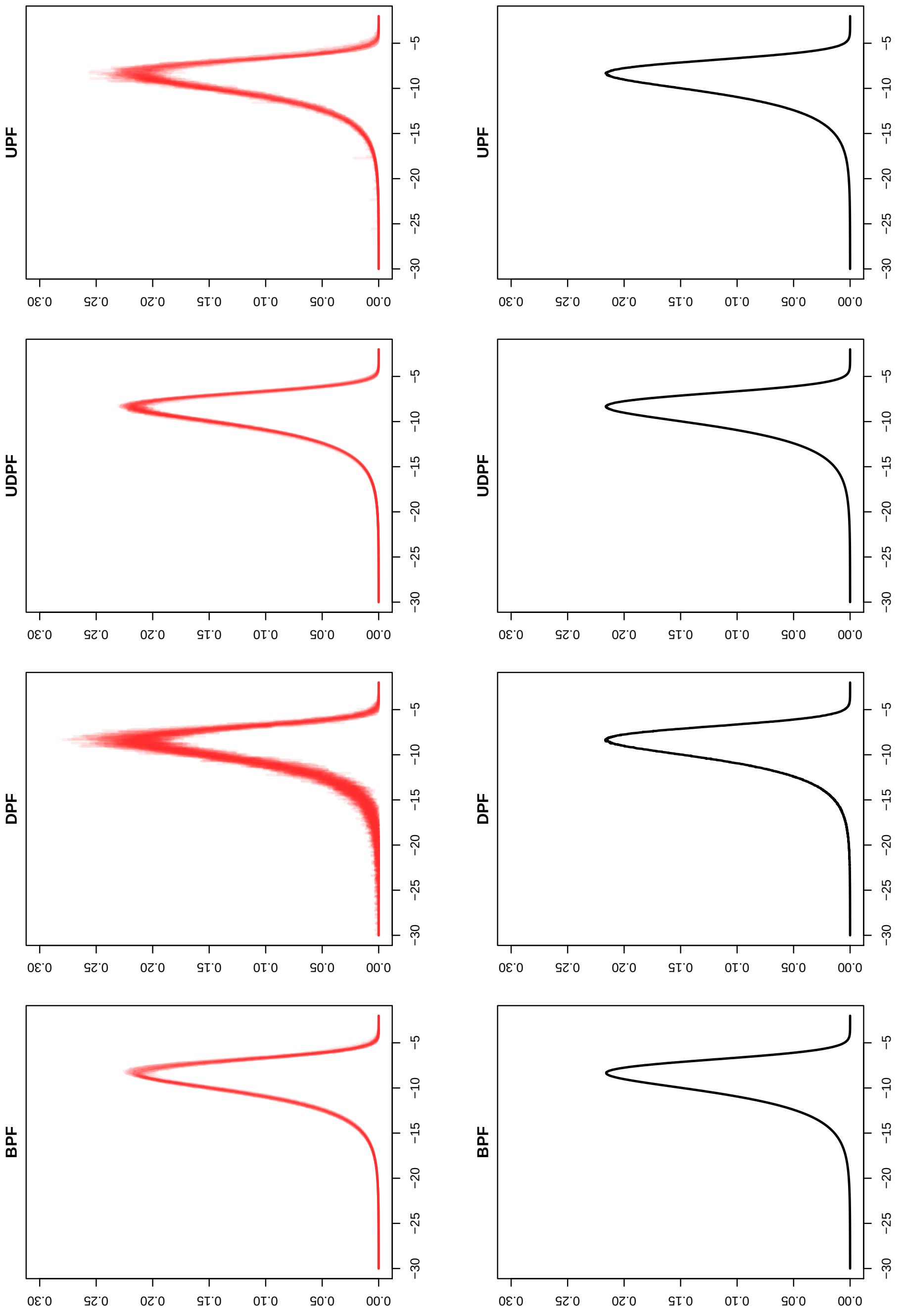

Figure 1: The top panel shows all of the individual *conditional* one-step ahead forecast distributions of $\log(y_{501}^2)$ produced by BPF, DPF, UDPF and UPF, for the correctly specified SV model, using simulated data. The bottom panel displays, for each filter, the *marginal* one-step ahead forecast distribution, which is obtained using the average of the corresponding conditional forecasts shown in the top panel.

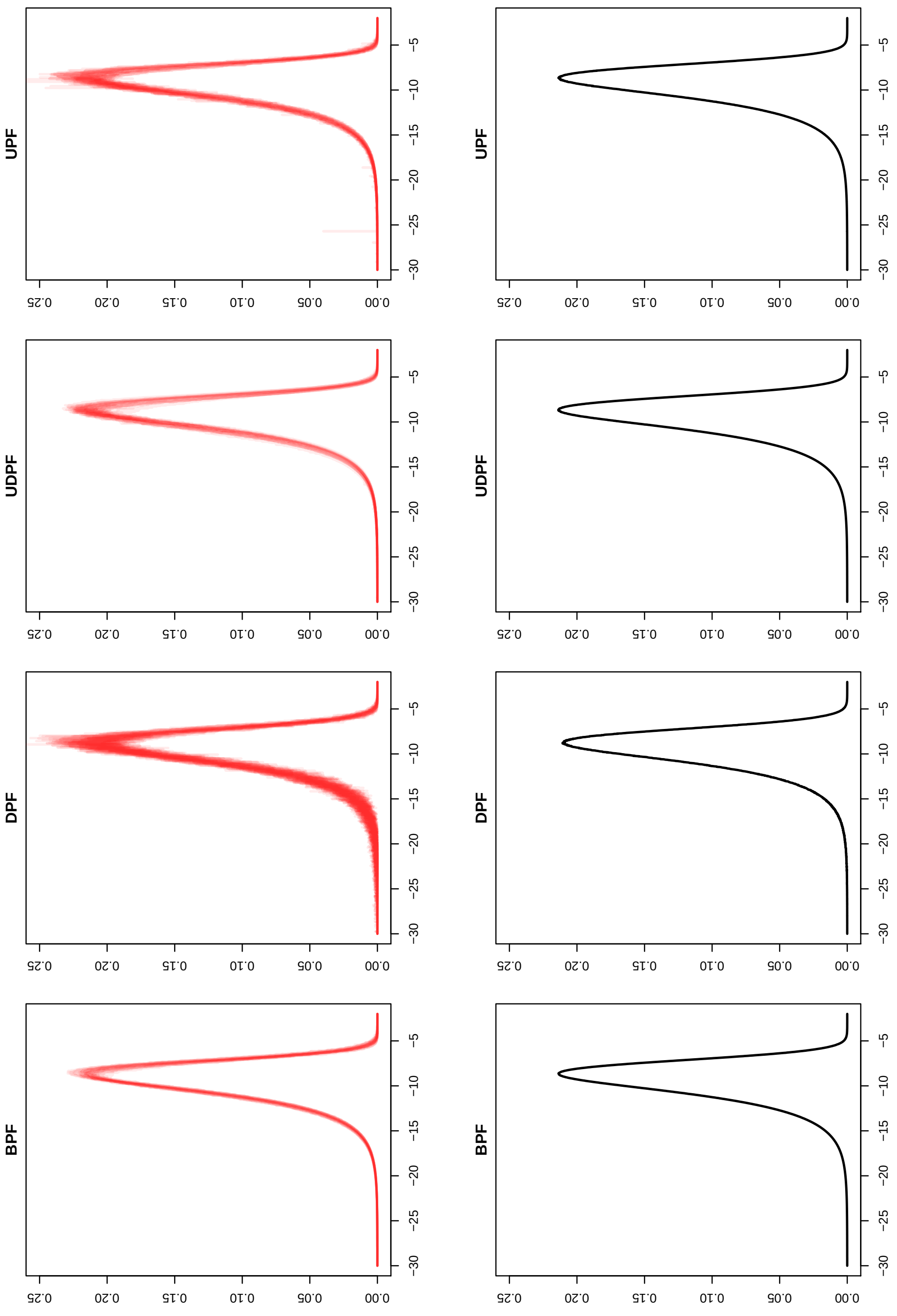

Figure 2: The top panel shows all of the individual *conditional* one-step ahead forecast distributions of $\log(y_{501}^2)$ produced by BPF, DPF, UDPF and UPF, for the incorrectly specified SV model, using simulated data. The bottom panel displays, for each filter, the *marginal* one-step ahead forecast distribution, which is obtained using the average of the corresponding conditional forecasts shown in the top panel.

Table 7: ALS (column 2) and ADLS (column 3) resulting from 254 one-step ahead out of sample forecasts obtained from the SV model estimated using PMMH, with four different filters.

Log score summaries		
	ALS	ADLS
DPF	-2.2447	0.0342
BPF	-2.2389	0.0000
UPF	-2.2471	0.0231
UDPF	-2.2381	0.0042

step-ahead predictive distributions. With each out-of-sample predictive and corresponding to each filter, a log score is produced. The corresponding forecast performance for the predictive distributions produced using the different filtering methods, as measured by ALS and ADLS, are reported in Table 7. These results show that the empirical forecast accuracy yielded by the distinct filters is almost identical, confirming the robustness of forecast performance to filter type documented above using simulation.

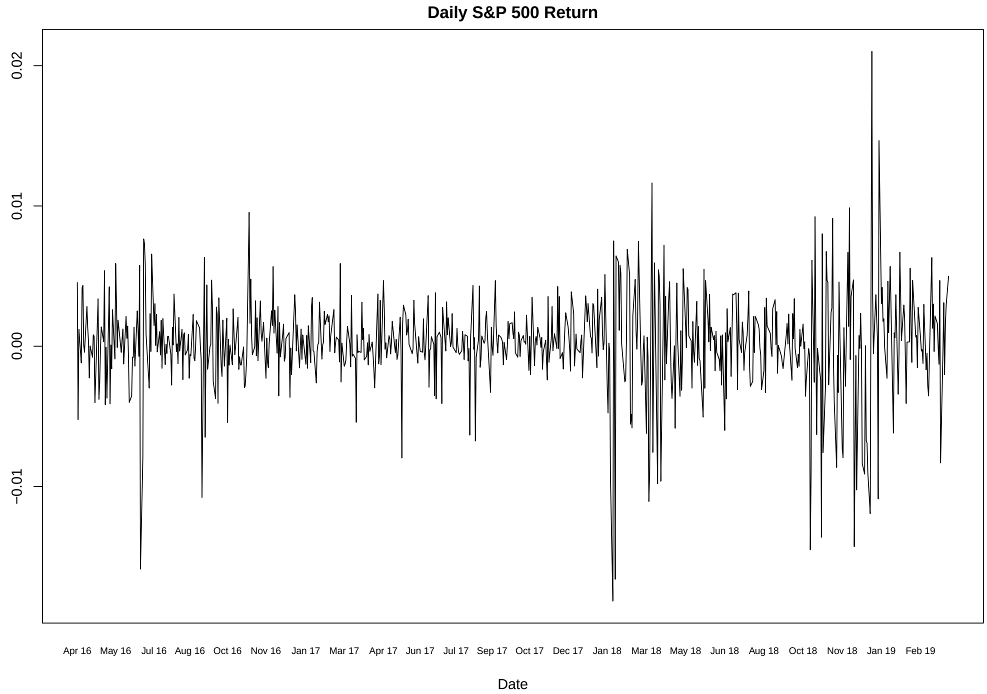

Figure 3: Time series plot of the 754 daily continuously compounded S&P 500 returns from 6-Apr-2016 to 2-Apr-2019.

6 Concluding remarks

This paper assesses the impact of filter choice on forecast accuracy in state space models in which PMMH algorithms are used to estimate the predictive distribution. To broaden the scope of the investigation, we complement certain well-established particle filters with two new particle filtering algorithms, namely the data-driven particle filter (DPF) and the unscented data-driven particle filter (UDPF), each of which is shown to yield an unbiased estimator of the likelihood function. In the context of several simulation studies and an empirical illustration, we show that out-of-sample forecast performance is largely insensitive to the choice of filtering method employed within the PMMH algorithm, and for the purpose of undertaking the forward filtering step. This finding holds irrespective of the correctness or otherwise of the model specification, including in an empirical setting, where it is inevitable that some degree of misspecification will prevail.

Of course, if parameter inference were the main focus, the choice of the filter may matter, with the adapted UDPF shown to perform well across a range of SNR settings (relative to the BPF, DPF and UPF), and to be on par with the fully adapted auxiliary particle filter (FAPF) when the latter is available. However, these and other caveats that may apply to the use of particle filtering within a Bayesian *inferential* algorithm, do not appear to apply to forecast performance. Our results suggest that when it comes to forecasting, subject to the qualification that a filter exhibits acceptably stable performance, the most appropriate decision is simply to use the most convenient filter for the model at hand.

A Appendices

A.1 Algorithms for implementing the DPF and UDPF

Algorithm 1 *The DPF with a pre-specified number of matches L, with* $1 \leq L \leq N$.

```
Generate $x_0^{[j]}$ from the initial state distribution $p(x_0)$, for $j = 1, 2, ..., N$.
Set the normalized particle weights $\pi_0^{[j]} = \frac{1}{N}, forj = 1, 2, ..., N$.
for $t = 0, 1, ..., T-1$ :
    Generate $\eta_{t+1}^{[j]} \overset{i.i.d.}{\sim} p(\eta_{t+1})$, for $j = 1, 2, ..., N$.
    Calculate the particle weights $w_{t+1}^{[j]}$ according to (17), for $j = 1, 2, ..., N$. Note that when
        $L = 1$ this is equivalent to (14).
    Calculate $\hat{p}_u(y_{t+1}|y_{1:t}, \theta)$ using (16).
    Calculate the normalized particle weights $\pi_{t+1}^{[j]} = \frac{w_{t+1}^{[j]}}{\sum_{k=1}^{N} w_{t+1}^{[k]}}$, for $j = 1, 2, ..., N$.
    Resample $N$ particles $x_{t+1}^{[j]}$, using probabilities $\{\pi_{t+1}^{[k]}, k = 1, 2, ..., N\}$.
    Set $\pi_{t+1}^{[j]} = \frac{1}{N}$.
```

Algorithm 2 *The UDPF.*

```
Generate $x_0^{[j]}$ from the initial state distribution $p(x_0)$, for $j = 1, 2, ..., N$.
Set the normalized particle weights $\pi_0^{[j]} = \frac{1}{N}, j = 1, 2, .., N$.
for $t = 0, 1, ..., T-1$ :
    Calculate $\hat{\mu}_{M,t+1}$ and $\hat{\sigma}_{M,t+1}^2$ according to (A.1) and (A.2).
    Construct the UDPF proposal distribution using (20), for $j = 1, 2, ..., N$.
```

6. Generate $x_{t+1}^{[j]}$ from the proposal distribution in Step 5, for $j = 1, 2, ..., N$.

7. Calculate the particle weights $w_{t+1}^{[j]}, j = 1, 2, .., N$ according to (21).

8. Calculate $\hat{p}_u(y_{t+1}|y_{1:t}, \theta)$ using (16).

9. Calculate the normalized particle weights $\pi_{t+1}^{[j]} = \frac{w_{t+1}^{[j]}}{\sum_{i=1}^{N} w_{t+1}^{[j]}}$, for $j = 1, 2, ..., N$.

10. Resample N particles $x_{t+1}^{[j]}$, using probabilities $\{\pi_{t+1}^{[k]}, k = 1, 2, ..., N\}$.

11. Set $\pi_{t+1}^{[j]} = \frac{1}{N}$.

A.2 The use of the unscented transformations in the UDPF

An unscented transformation is a quick and accurate procedure for calculating the moments of a non-linear transformation of an underlying random variable. The procedure involves choosing a set of points, called *sigma points*, from the support of the underlying random variable. Once selected, these sigma points are weighted to ensure that the first $M - 1$ moments of the discrete sigma point distribution equal the first $M-1$ moments of the corresponding distribution of the underlying random variable. The set of sigma points is then propagated through the relevant non-linear function, from which the mean and variance of the resulting normal approximation are obtained. The implied moments associated with the weighted transformed points can be shown to match the true moments of the transformed underlying random variable up to a predetermined order of accuracy. (See Julier, Uhlmann and Durrant-Whyte, 1995, 2000.)

In the UDPF, the unscented transformation is applied to the function defined by solving the measurement equation in (10) for the state variable x_{t+1}. We denote μ_η and σ_η^2 respectively as the expected value and the variance of the measurement error η_{t+1}. To calculate the mean and variance of the normal approximation in (18), sigma points $\eta^{\{k\}}$, with $k = 1, ..., M$, are chosen to span the support of η_{t+1}. The corresponding weights for each each sigma point, $Q^{\{k\}}$, are determined to ensure that the first $M - 1$ moments of the (discrete) distribution associated with the weighted sigma points match the corresponding theoretical moments of the underlying distribution, $p(\eta_{t+1})$. Accordingly, the sigma point weights satisfy the following system of equations

$$\left\{ \begin{array}{c} \sum_{k=1}^{M} Q^{\{k\}} \\ \sum_{k=1}^{M} Q^{\{k\}}(\eta^{\{k\}} - \mu_\eta) \\ \vdots \\ \sum_{k=1}^{M} Q^{\{k\}}(\eta^{\{k\}} - \mu_\eta)^{M-1} \end{array} \right. = \begin{array}{c} 1 \\ E\left[\eta_{t+1} - \mu_\eta\right] \\ \vdots \\ E\left[(\eta_{t+1} - \mu_\eta)^{M-1}\right] \end{array} .$$

Note that, if the measurement errors have the same distribution for all t, then the weighted sigma point distribution will also be the same for all t, and hence will require calculation only once. This is the situation for all models considered in the paper.

For implementation of the unscented transformations within the UPDF, let the mean of the distribution whose density is proportional to the measurement density be given by

$$\mu_{M,t+1} = \int_{-\infty}^{\infty} x_{t+1} C_{t+1} p(y_{t+1}|x_{t+1}, \theta) dx_{t+1},$$

where $C_{t+1} = (\int p(y_{t+1}|x_{t+1}, \theta) dx_{t+1})^{-1}$ represents the normalizing constant that ensures a proper density. Further, using the representation of the measurement density in (11), we have

$$\mu_{M,t+1} = \int_{-\infty}^{\infty} x_{t+1} C_{t+1} \int_{-\infty}^{\infty} p(\eta_{t+1}) \left| \frac{\partial h}{\partial x_{t+1}} \right|_{x_{t+1}=x_{t+1}(y_{t+1},\eta_{t+1})}^{-1} \delta_{x_{t+1}(y_{t+1},\eta_{t+1})} d\eta_{t+1} dx_{t+1}.$$

Using then the discrete approximation of $p\left(\eta_{t+1}\right)$ implied by the weighted sigma points, $\eta^{\{k\}}$ for $k = 1, 2, ..., M$, the mean of the measurement component as calculated by the unscented transformation satisfies

$$\begin{aligned}
\widehat{\mu}_{M,t+1} &= \int_{-\infty}^{\infty} x_{t+1} C_{t+1} \int_{-\infty}^{\infty} \widehat{p}(\eta_{t+1}) \left| \frac{\partial h}{\partial x_{t+1}} \right|_{x_{t+1}=x_{t+1}(y_{t+1},\eta_{t+1})}^{-1} \delta_{x_{t+1}(y_{t+1},\eta_{t+1})} d\eta_{t+1} dx_{t+1} \\
&= \int_{-\infty}^{\infty} x_{t+1} C_{t+1} \int_{-\infty}^{\infty} \left[\sum_{k=1}^{M} Q^{\{k\}} \delta_{\eta^{\{k\}}} \right] \left| \frac{\partial h}{\partial x_{t+1}} \right|_{x_{t+1}=x_{t+1}(y_{t+1},\eta_{t+1})}^{-1} \delta_{x_{t+1}(y_{t+1},\eta_{t+1})} d\eta_{t+1} dx_{t+1} \\
&= \frac{\sum_{k=1}^{M} Q^{\{k\}} \left| \frac{\partial h}{\partial x_{t+1}} \right|_{\eta_{t+1}=\eta^{\{k\}},\ x_{t+1}=x_{t+1}(y_{t+1},\eta^{\{k\}})}^{-1} x_{t+1}(y_{t+1}, \eta^{\{k\}})}{\sum_{j=1}^{M} Q^{\{j\}} \left| \frac{\partial h}{\partial x_{t+1}} \right|_{\eta_{t+1}=\eta^{\{j\}},\ x_{t+1}=x_{t+1}(y_{t+1},\eta^{\{j\}})}^{-1}}. \qquad \text{(A.1)}
\end{aligned}$$

Similarly, the variance of the measurement component as calculated by the unscented transformation is given by

$$\widehat{\sigma}_{M,t+1}^2 = \frac{\sum_{k=1}^{M} Q^{\{k\}} \left| \frac{\partial h}{\partial x_{t+1}} \right|_{\eta_{t+1}=\eta^{\{k\}},\ x_{t+1}=x_{t+1}(y_{t+1},\eta^{\{k\}})}^{-1} (x_{t+1}(y_{t+1}, \eta^{\{k\}}) - \widehat{\mu}_{M,t+1})^2}{\sum_{j=1}^{M} Q^{\{k\}} \left| \frac{\partial h}{\partial x_{t+1}} \right|_{\eta_{t+1}=\eta^{\{j\}},\ x_{t+1}=x_{t+1}(y_{t+1},\eta^{\{j\}})}^{-1}}. \qquad \text{(A.2)}$$

A.3 The unbiasedness of the data-driven filters

As discussed in Section 2.1, the unbiasedness condition in (4) is required to ensure that a PMMH scheme yields the correct invariant posterior distribution for θ, and so we consider the theoretical properties of the new filters proposed in Sections 3.2.1 and 3.2.2 here. While the proof in Pitt *et al.* (2012) demonstrates the unbiasedness property of the likelihood estimator produced by the APF, their proof also applies for the BPF and UPF. That the proof applies to the BPF is noted by Pitt *et al.*, with the critical insight being that the first step of their Algorithm 1 has no impact, so that each previous particle $x_t^{[j]}$ retains the weight of $\pi_t^{[j]} = 1/N$. This is also true of the UPF, as the information regarding the next observation y_{t+1} is incorporated into the proposal distribution at time $t + 1$ through an unscented transformation, and not via an additional resampling step.

However, due to the multiple matching technique, which is only available for use with the DPF (and not with either the APF or the UPF), the proof in Pitt *et al.* is not adequate to prove Theorem 1. In this case we provide all details of the proof of Theorem 1 here, along with those of two lemmas upon which our proof depends. In particular, the following two theorems establish that the unbiasedness condition holds for all versions of the data-driven filter, namely the DPF with single ($L = 1$), partial ($L < N$) or full ($L = N$) matching. The collective conditions C1 - C3 detailed below, which ensure that the outlined algorithms produce well-defined proposal distributions, are assumed when deriving the unbiasedness of the resulting likelihood estimators.

C1. For each fixed value x, the function $h(x, \eta)$ is a strictly monotone function of η, with continuous non-zero (partial) derivative.

C2. For each fixed value y, the function $x(y, \eta)$, defined implicitly by $y = h(x, \eta)$, is a strictly monotone function of η, with continuous non-zero (partial) derivative.

C3. The conditions $\int x_{t+1}^k p(y_{t+1}|x_{t+1}, \theta) dx_{t+1} < \infty$ hold, for $k = 0$, 1, and 2.

Theorem 1 *Under C1 through C2, any likelihood estimator produced by a DPF is unbiased. That is, the likelihood estimator* $\widehat{p}_u(y_{1:T}|\theta)$ *resulting from any such filter, with* $1 \le L \le N$ *matches, satisfies*

$$E_u[\widehat{p}_u(y_{1:T}|\theta)] = p(y_{1:T}|\theta).$$

We adapt the proof from Pitt *et al.* (2012) to demonstrate the unbiasedness of the new likelihood estimators specified under Theorem 1, and represented generically by

$$\widehat{p}_u(y_{1:T}|\theta) = \widehat{p}_u(y_1|\theta) \prod_{t=2}^{T} \widehat{p}_u(y_t|y_{1:t-1}, \theta), \tag{A.3}$$

where unbiasedness means that $E[\widehat{p}_u(y_{1:T})|\theta] = p(y_{1:T}|\theta)$. The factors in (A.3) are given in (16) for each $t = 1, 2, ..., T$, with the weights $w_{t+1}^{[j]}$ defined by the relevant choice of L. As conditioning on the parameter θ remains in all subsequent expressions, we again suppress its explicit inclusion to help simplify the expressions throughout the remainder of this appendix.

Firstly, as noted above, the conditions outlined for this theorem ensure that the proposal distribution, $g(x_{t+1}|y_{t+1}, \theta)$, is well defined. Next, let u denote the vector of canonical *i.i.d.* random variables used to implement the given filtering algorithm, and let F_t be the subset of such variables generated up to and including time t, for each $t = 0, 1, ..., T$. This means that by conditioning on F_t, the particle set $\left\{x_{0:t}^{[1]}, x_{0:t}^{[2]}, ..., x_{0:t}^{[N]}\right\}$ and the associated normalized weights $\left\{\pi_t^{[1]}, \pi_t^{[2]}, ..., \pi_t^{[N]}\right\}$ that together provide the approximation of the filtered density, as in (9), are assumed to be known. Following Pitt *et al.* (2012), in order to prove the unbiasedness property of the likelihood estimator we require the following two lemmas:

Lemma 1

$$E_u[\widehat{p}_u(y_T|y_{1:T-1}, \theta)|F_{T-1}] = \sum_{j=1}^{N} \pi_{T-1}^{[j]} p(y_T|x_{T-1}^{[j]}, \theta).$$

Lemma 2

$$E_u[\widehat{p}_u(y_{T-h:T}|y_{1:T-h-1}, \theta)|F_{T-h-1}] = \sum_{j=1}^{N} \pi_{T-h-1}^{[j]} p(y_{T-h:T}|x_{T-h-1}^{[j]}, \theta). \tag{A.4}$$

According to Section 3.2.1, the estimator of the likelihood component for the DPF (with potential multiple matching), for given $1 \leq L \leq N$, is

$$\begin{aligned}
\widehat{p}_u(y_{t+1}|y_{1:t},\theta) &= \sum_{j=1}^{N} w_{t+1}^{[j]} \\
&= \sum_{j=1}^{N}\left(\frac{1}{L}\sum_{l=1}^{L} w_{t+1}^{[j][l]}\right) \\
&= \sum_{j=1}^{N}\left(\frac{1}{L}\sum_{l=1}^{L} \frac{p(y_{t+1}|x_{t+1}^{[j]},\theta)\pi_t^{[k_{l,j}]}p(x_{t+1}^{[j]}|x_t^{[k_{l,j}]},\theta)}{g(x_{t+1}^{[j]}|y_{t+1},\theta)}\right), \qquad \text{(A.5)}
\end{aligned}$$

where the proposal distribution is given in (12) and $k_{l,j}$ represents the j^{th} component of the l^{th} cyclic permutation, K_l, as defined in Section 3.2.1.

Proof of Lemma 1. We start with

$$\begin{aligned}
E_u[\widehat{p}_u(y_T|y_{1:T-1},\theta)|F_{T-1}] &= E_u\left[\sum_{j=1}^{N}\left(\frac{1}{L}\sum_{l=1}^{L}\frac{p(y_T|x_T^{[j]},\theta)\pi_{T-1}^{[k_{l,j}]}p(x_T^{[j]}|x_{T-1}^{[k_{l,j}]},\theta)}{g(x_T^{[j]}|y_T,\theta)}\right)\middle| F_{T-1}\right] \\
&= \frac{1}{L}\sum_{j=1}^{N}\sum_{l=1}^{L} E_u\left[\pi_{T-1}^{[k_{l,j}]}\frac{p(x_T^{[j]}|x_{T-1}^{[k_{l,j}]},\theta)p(y_T|x_T^{[j]},\theta)}{g(x_T^{[j]}|y_T,\theta)}\middle| F_{T-1}\right].
\end{aligned}$$

The randomness of each component within the double summation, for which the expectation is to be taken, comes from the proposal distribution that simulates the particle $x_T^{[j]}$. Hence, the expectation can be replaced with its integral form explicitly as:

$$\begin{aligned}
E_u[\widehat{p}_u(y_T|y_{1:T-1},\theta)|F_{T-1}] &= \frac{1}{L}\sum_{j=1}^{N}\sum_{l=1}^{L}\int \pi_{T-1}^{([k_{l,j}]}\frac{p(x_T|x_{T-1}^{[k_{l,j}]},\theta)p(y_T|x_T,\theta)}{g(x_T|y_T,\theta)}g(x_T|y_T,\theta)dx_T \qquad \text{(A.6)} \\
&= \frac{1}{L}\sum_{j=1}^{N}\sum_{l=1}^{L}\left\{\pi_{T-1}^{[k_{l,j}]}\int p(y_T,x_T|x_{T-1}^{[k_{l,j}]},\theta)dx_T\right\} \\
&= \frac{1}{L}\sum_{j=1}^{N}\sum_{l=1}^{L}\pi_{T-1}^{[k_{l,j}]}p(y_T|x_{T-1}^{[k_{l,j}]},\theta).
\end{aligned}$$

Since the N permutations of the previous particles are mutually exclusive, each of the terms within the double summation appears exactly L times. Therefore,

$$\begin{aligned}
E_u[\widehat{p}_u(y_T|y_{1:T-1},\theta)|F_{T-1}] &= \frac{1}{L}\sum_{j=1}^{N} L\left[\pi_{T-1}^{[j]}p(y_T|x_{T-1}^{[j]},\theta)\right] \\
&= \sum_{j=1}^{N}\pi_{T-1}^{[j]}p(y_T|x_{T-1}^{[j]},\theta).
\end{aligned}$$

Hence, Lemma 1 holds. ■

Proof of Lemma 2. To prove Lemma 2, we use method of induction as per Pitt *et al.* (2012) First note that, according to Lemma 1, (A.4) holds when $h = 0$. Next, assuming that Lemma 2 holds for any integer $h \geq 0$, we show that it also holds for $h + 1$.

By the law of iterated expectations, we have

$$
\begin{aligned}
&E_u[\widehat{p}_u(y_{T-h-1:T}|y_{1:T-h-2},\theta)|F_{T-h-2}] \\
&= E_u\left[E_u\left[\widehat{p}_u(y_{T-h:T}|y_{1:T-h-1},\theta)|F_{T-h-1}\right]\widehat{p}_u(y_{T-h-1}|y_{1:T-h-2},\theta)|F_{T-h-2}\right].
\end{aligned}
$$

By substituting the formula of $\widehat{p}_u(y_{T-h-1}|\theta, y_{1:T-h-2})$ and using the assumption that Lemma 2 holds for h, we have

$$
\begin{aligned}
&E_u[\widehat{p}_u(y_{T-h-1:T}|y_{1:T-h-2},\theta)|F_{T-h-2}] \\
&= E_u\left[\left\{\sum_{j=1}^{N}\pi_{T-h-1}^{[j]}p(y_{T-h:T}|x_{T-h-1}^{[j]},\theta)\right\}\left\{\sum_{j=1}^{N}w_{T-h-1}^{[j]}\right\}\middle| F_{T-h-2}\right]
\end{aligned}
$$

and noting that $\pi_{T-h-1}^{[j]}$ is the normalized version of $w_{T-h-1}^{[j]}$, then

$$
\begin{aligned}
&E_u[\widehat{p}_u(y_{T-h-1:T}|y_{1:T-h-2},\theta)|F_{T-h-2}] \\
&= E_u\left[\left\{\frac{\sum_{j=1}^{N}p(y_{T-h:T}|x_{T-h-1}^{[j]},\theta)w_{T-h-1}^{[j]}}{\sum_{k=1}^{N}w_{T-h-1}^{[k]}}\right\}\left\{\sum_{j=1}^{N}w_{T-h-1}^{[j]}\right\}\middle| F_{T-h-2}\right] \\
&= \sum_{j=1}^{N}E_u\left[p(y_{T-h:T}|x_{T-h-1}^{[j]},\theta)w_{T-h-1}^{[j]}\middle| F_{T-h-2}\right].
\end{aligned}
$$

Adopting a similar procedure to that above, owing to the fact that the expectation is taken with respect to the relevant proposal distribution and that the multiple matches employ only cyclic rotations, we have

$$
\begin{aligned}
&E_u\left[\widehat{p}_u(y_{T-h-1:T}|y_{1:T-h-2},\theta)|\, F_{T-h-2}\right] \\
&= \sum_{j=1}^{N}E_u\left[p(y_{T-h:T}|x_{T-h-1}^{[j]},\theta)\frac{p(y_{T-h-1}|x_{T-h-1}^{[j]},\theta)\frac{1}{L}\sum_{l=1}^{L}\pi_{T-h-2}^{[k_{l,j}]}p(x_{T-h-1}^{[j]}|x_{T-h-2}^{[k_{l,j}]},\theta)}{g(x_{T-h-1}^{[j]}|y_{T-h-1},\theta)}\middle| F_{T-h-2}\right] \\
&= \frac{1}{L}\sum_{j=1}^{N}\sum_{l=1}^{L}\pi_{T-h-2}^{[k_{l,j}]}\int p(y_{T-h:T}|x_{T-h-1},\theta)p(y_{T-h-1}|x_{T-h-1},\theta)p(x_{T-h-1}|x_{T-h-2}^{[k_{l,j}]},\theta)dx_{T-h-1} \\
&= \sum_{j=1}^{N}\left\{\pi_{T-h-2}^{[j]}\int p(y_{T-h:T}|x_{T-h-1},\theta)p(y_{T-h-1}|x_{T-h-1},\theta)p(x_{T-h-1}|x_{T-h-2}^{[j]},\theta)dx_{T-h-1}\right\} \\
&= \sum_{j=1}^{N}\pi_{T-h-2}^{[j]}p(y_{T-h-1:T}|x_{T-h-2}^{[j]},\theta)
\end{aligned}
$$

as required. ■

Proof of Theorem 1. From Lemma 2, when $h = T-1$, then

$$
E_u\left[\widehat{p}_u(y_{1:T}|\theta)|\, F_0\right] = \sum_{j=1}^{N}p(y_{1:T}|x_0^{[j]},\theta)\pi_0^{[j]}.
$$

Next, marginalizing over the randomness of u associated with generating a set of equally weighted particles, $\left\{x_0^{[1]}, x_0^{[2]}, ..., x_0^{[N]}\right\}$ at time $t = 0$ from the initial distribution $p(x_0)$, we have

$$
\begin{aligned}
E_u\left[\widehat{p}_u(y_{1:T}|\theta)\right] &= E_u\left[E_u\left[\widehat{p}_u(y_{1:T}|\theta)|\, F_0\right]\right] \\
&= E_u\left[\sum_{j=1}^{N} p(y_{1:T}|x_0^{[j]}, \theta)\pi_0^{[j]}\right] \\
&= \frac{1}{N}\sum_{j=1}^{N} E_u\left[p(y_{1:T}|x_0^{[j]}, \theta)\right].
\end{aligned}
$$

Finally, since the expectation of $p(y_{1:T}|x_0^{[j]}, \theta)$ is the same for all j, then

$$
\begin{aligned}
E_u\left[\widehat{p}_u(y_{1:T}|\theta)\right] &= E_u\left[p(y_{1:T}|x_0, \theta)\right] \\
&= \int p(y_{1:T}|x_0, \theta)p(x_0|\theta)dx_0 \\
&= p(y_{1:T}|\theta),
\end{aligned}
$$

and the unbiasedness property of the likelihood estimator associated with each of the DPF algorithms specified under Theorem 1 is established. ■

Theorem 2 *Under C1 through C3, the likelihood estimator produced by the UDPF filter is unbiased.*

Proof of Theorem 2. By recognizing the similarity between the UDPF and the APF, the proof of Theorem 2 can be deduced directly from the unbiasedness proof of Pitt *et al.* (2012). In reference to Algorithm 1 of Pitt *et al.*, the UDPF algorithm can be reconstructed by setting $g(y_{t+1}|x_t^{[j]}, \theta) = 1$ and with the proposal distribution, $g(x_{t+1}|x_t^{[j]}, y_{t+1}, \theta)$, formed as per (20). All that is required is that we ensure, through sufficient conditions C1 - C3, that the approximate moments $\widehat{\mu}_{M,t+1}$ and $\widehat{\sigma}^2_{M,t+1}$ used to obtain this Gaussian proposal distribution are finite. ■

References

[1] Andrieu, C., Doucet, A. and Holenstein, R. 2010. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3), pp. 269-342.

[2] Andrieu, C. and Roberts, G. 2009. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2), pp. 697-725.

[3] Au, C. and Tam, J. 1999. Transforming variables using the Dirac generalized function. *The American Statistician* 53, 270-272.

[4] Beaumont, M. A. 2003. Estimation of population growth or decline in genetically monitored populations. *Genetics* 164, 1139–1160.

[5] Bauwens, L. and Veredas, D. 2004. The stochastic conditional duration model: a latent variable model for the analysis of financial durations. *Journal of Econometrics*, 119(2), pp. 381-412.

[6] Chopin, N. and Singh S.S. 2015. On particle Gibbs sampling. *Bernoulli*, 21(3), pp. 1855–1883.

[7] Creal, D., 2012. A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews*, 31(3), pp. 245-296.

[8] Deligiannidis, G., Doucet, A. and Pitt, M.K. 2018. The correlated pseudomarginal method, *Journal of the Royal Statistical Society, Series B,* 80, 839-870

[9] Del Moral, P. 2004. *Feynman-Kac formulae: Genealogical and Interacting Particle Systems with Applications*. New York, Springer Verlag.

[10] Del Moral, P., Jasra, A., Lee, A., Yau, C. and Zhang, X. 2015. The alive particle filter and its use in particle Markov chain Monte Carlo. *Stochastic Analysis and Applications*, 33(6), pp. 943-974.

[11] Del Moral, P. and Murray, L.M. 2015. Sequential Monte Carlo with highly informative observations. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1), pp. 969-997.

[12] Doucet, A. and Johansen, A.M. 2011. A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan, D., and Rozovskii, B. (eds.) *The Oxford Handbook of Nonlinear Filtering*, New York, Oxford University Press.

[13] Doucet, A. and Lee, A. 2018. Sequential Monte Carlo Methods. *Handbook of Graphical Models,* Eds M. Maathuis. M. Drton, S. Lauritzen and M. Wainwright, Chapman and Hall/CRC.

[14] Doucet, A., Pitt, M.K., Deligiannidis, G. and Kohn, R. 2015. Efficient implementation of Markov chain Monte Carlo, *Biometrika*, 102, 295-313.

[15] Duffie D, Pan J, Singleton K. 2000. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68: 1343–1376.

[16] Flury, T. and Shephard, N. 2011. Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory*, 27(5), pp. 933-956.

[17] Fox, D., Thrun, S., Burgard, W., and Dellaert, F. 2001. Particle filters for mobile robot localization. In: Doucet, A., de Freitas, N., and Gordon, N. (eds.) *Sequential Monte Carlo Methods in Practice*. New York, Springer Verlag.

[18] Frazier, D. T., Maneesoonthorn, W., Martin, G. M., and McCabe, B. P. 2019. Approximate Bayesian forecasting. *International Journal of Forecasting*, 35(2):521–539.

[19] Geweke, J. and Amisano, G. 2010. Comparing and evaluating Bayesian predictive distributions of asset returns. *International Journal of Forecasting*, 26, pp. 216-230.

[20] Giordani, P., Pitt, M.K., and Kohn, R. 2011. Bayesian inference for time series state space Models. In: Geweke, J., Koop, G., and van Dijk, H. (eds.) *The Oxford Handbook of Bayesian Econometric*s, New York, Oxford University Press.

[21] Gordon, N.J., Salmond, D.J. and Smith, A.F. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F,* 140(2), pp. 107-113.

[22] Guarniero, P., Johansen, A.M. and Lee, A. 2017. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112, 1636-1647.

[23] Hannig, J. Iyer, H., Lai, R.C.S. and Lee, T.C.M. (2016). Generalized fiducial inference: a review and new results. *Journal of the American Statistical Association*, 111(515), pp. 1346-1361.

[24] Julier, S.J. and Uhlmann, J.K. 1997. New extension of the Kalman filter to nonlinear systems. *Proceedings Signal Processing, Sensor Fusion, and Target Recognition* VI, 3068, pp. 182-193.

[25] Julier, S. J., Uhlmann, J.K. and Durrant-Whyte, H.F. 1995. A new approach for filtering nonlinear systems. *Proceedings of the 1995 American Control Conference*, 3, pp. 1628-1632.

[26] Julier, S. J., Uhlmann, J.K. and Durrant-Whyte, H.F. 2000. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control,* 45(3)*,* pp. 477-482.

[27] Klaas, M., de Freitas, N. and Doucet, A. 2012. Toward practical N2 Monte Carlo: the marginal particle filter. *arXiv preprint arXiv:1207.1396.*

[28] Lin, M.T., Zhang, J.L., Cheng, Q. and Chen, R. 2005. Independent particle filters. *Journal of the American Statistical Association*, 100(472), pp. 1412-1421.

[29] Lin, M., Chen, R. and Liu, J.S. 2013. Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28(1), pp. 69-94.

[30] Lindsten, F., Jordan, M.I. and Schön, T.B. 2014. Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research* 15, pp. 2145-2184.

[31] Maneesoonthorn, W., Forbes, C.S. and Martin, G.M. 2017. Inference on self-exciting jumps in prices and volatility using high-frequency measures. *Journal of Applied Econometrics*, 32, pp. 504-532.

[32] Müller, C.L. 2010. Exploring the common concepts of adaptive MCMC and Covariance Matrix Adaptation schemes. *Dagstuhl Seminar Proceedings.* Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[33] Ng, J., Forbes, C.S., Martin, G.M. and McCabe, B.P. 2013. Non-parametric estimation of forecast distributions in non-Gaussian, non-linear state space models. *International Journal of Forecasting*, 29(3), pp. 411-430.

[34] Pitt, M.K. and Shephard, N. 1999. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446), pp. 590-599.

[35] Pitt, M.K., dos Santos Silva, R., Giordani, P. and Kohn, R. 2012. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2), pp. 134-151.

[36] Quiroz, M., Tran, M-N. Villani V. and Kohn, R. 2018. Speeding up MCMC by delayed acceptance and data subsampling, *Journal of Computational and Graphical Statistics*, 27, 12-22.

[37] Quiroz, M., Nott, D. J., and Kohn, R. 2018. Gaussian variational approximation for high-dimensional state space models. *https://arXiv:1801.07873.*

[38] Quiroz, M., Tran, M-N. Villani V. and Kohn, R. 2019. Speeding up MCMC by efficient data subsampling, *Journal of the American Statistical Association*, 114(526), pp. 831-843.

[39] Shephard, N. 2005. *Stochastic Volatility: Selected Readings.* New York, Oxford University Press.

[40] Smith, J.Q. and Santos, A.A.F. 2006. Second-order filter distribution approximations for financial time series with extreme outliers. *Journal of Business & Economic Statistics*, 24(3), pp. 329-337.

[41] Strickland, C.M. Forbes, C.S. and Martin, G.M. 2006. Bayesian analysis of the stochastic conditional duration model, *Computational Statistics and Data Analysis, Special Issue on Statistical Signal Extraction and Filtering,* 50, pp. 2247-2267.

[42] Taylor, S.J. 1982. Financial returns modelled by the product of two stochastic processes, a study of daily sugar prices 1961-79. In: Anderson, O. D. (ed.), *Time Series Analysis: Theory and Practice 1*, North-Holland, Amsterdam.

[43] van de Merwe, R., Doucet, A., de Freitas, N., and Wan, E. 2000. The unscented particle filter, advances in neural information processing systems. Available at http://books.nips.cc/papers/files/nips13/MerweDoucetFreitasWan.pdf.

[44] Whiteley, N. and Lee, A. 2014. Twisted particle filters. *The Annals of Statistics*, 42(1), pp. 115-141.