Data-driven Particle Filters for Particle Markov Chain Monte Carlo*

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

December 21, 2016

Abstract

This paper proposes new automated proposal distributions for sequential Monte Carlo algorithms, including particle filtering and related sequential importance sampling methods. The weights for these proposal distributions are easily established, as is the unbiasedness property of the resultant likelihood estimators, so that the methods may be used within a particle Markov chain Monte Carlo (PMCMC) inferential setting. Simulation exercises, based on a range of state space models, are used to demonstrate the linkage between the signal-to-noise ratio of the system and the performance of the new particle filters, in comparison with existing filters. In particular, we demonstrate that one of our proposed filters performs well in a high signal-to-noise ratio setting, that is, when the observation is informative in identifying the location of the unobserved state. A second filter, deliberately designed to draw proposals that are informed by both the current observation and past states, is shown to work well across a range of signal-to-noise ratios and to be much more robust than the auxiliary particle filter, which is often used as the default choice. We then extend the study to explore the performance of the PMCMC algorithm using the new filters to estimate the likelihood function, once again in comparison with existing alternatives. Taking into consideration robustness to the signal-to-noise ratio, computation time and the efficiency of the chain, the second of the new filters is again found to be the best-performing method. Application of the preferred filter to a stochastic volatility model for weekly Australian/US exchange rate returns completes the paper.

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

1 INTRODUCTION

Bayesian inference using a likelihood function estimated via simulation has drawn increasing attention from researchers of late. This interest has been prompted by the seminal work by Andrieu,

*The authors would like to thank participants at the Melbourne Bayesian Econometrics Workshop, 2015; the Monash Workshop on Financial Econometrics, 2015; the Computational Financial Econometrics Conference, 2015; the International Association for Statistical Computing, Asian Regional Section Meeting, 2015; and the Econometric Society Australasian Meetings, 2016, for constructive feedback on the paper. The research has been supported by Australian Research Council Discovery Grant DP150101728.

†Department of Econometrics and Business Statistics, Monash University.

‡Department of Econometrics and Business Statistics, Monash University. Corresponding author; email: catherine.forbes@monash.edu; postal address: 20 Chancellors Walk, Monash University, Victoria 3800, Australia.

§Department of Econometrics and Business Statistics, Monash University.

¶Management School, University of Liverpool.

Doucet and Holenstein (2010), in which exact Bayesian inference is shown to be achievable via the insertion of an unbiased likelihood estimator within a Metropolis-Hastings (MH) Markov chain Monte Carlo (MCMC) algorithm. The unbiased likelihood estimators given focus therein are based on particle filters already established in the literature. Subsequent work by Flury and Shephard (2011) and Pitt, dos Santos Silva, Giordani and Kohn (2012) explores the interface between different filtering-based estimates of the likelihood and the mixing properties of the resultant particle Markov chain Monte Carlo (PMCMC) algorithm 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 analysis. (See also Whiteley and Lee, 2014, Del Moral, Jasra, Lee, Yau and Zhang, 2015, Del Moral and Murray, 2015, and Guarniero, Johansen and Lee, 2016.)

The focus of the current paper is on the development of a new family of independent particle filters (IPFs) and on the exploration of their performance in a PMCMC scenario. A generic IPF algorithm, whereby the proposal distribution is derived from the current observation only, and does not include past (forecasted) information about the current latent state variable, was first introduced by Fox, Thrun, Burgard and Dellaert (2001), Lin, Zhang, Cheng and Chen (2005) and Lin, Chen and Liu (2013). In contrast with the bootstrap particle filter (BPF) of Gordon, Salmond and Smith (1993) in particular, this form of algorithm has been shown to perform well in experimental settings when the system has a large signal-to-noise ratio; that is, when the current observation provides significant information about the location of the underlying state. However, no systematic method for obtaining proposal draws to implement an IPF has thus far been proposed, and no assessment of the (relative) performance of the filter in an inferential setting has been undertaken. These are gaps we look to fill.

Drawing on a novel 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 contribution of the paper is a modification of the 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. In contrast to the unscented particle filter (UPF) of van de Merwe, Doucet, de Freitas and Wan (2000) - which also makes use of such transformations - the

UDPF transforms moments of the measurement error using the unscented method. The computational benefit of using a multiple matching of particles (Lin *et al.*, 2005) in the production of the likelihood estimate is explored in the context of the DPF, and it is established that the likelihood estimators resulting from both new methods are unbiased.

The numerical performance of the DPF-based filters is compared with that of the BPF, the auxiliary particle filter (APF) of Pitt and Shephard (1999) and the UPF in a series of simulation experiments. The experiments are 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 object of interest is initially the likelihood function estimated at a given fixed parameter vector that accords with that underlying the true data generating process (DGP). This exercise allows the impact on the performance of the filters of the characteristics of the DGP - in particular the signal-to-noise ratio - to be documented, abstracting from the issue of parameter uncertainty. Accuracy of the likelihood estimate is assessed using the exact likelihood function, evaluated by the Kalman filter (KF) in the case of i) and by the (deterministic) non-linear grid-based filter of Ng *et al.* (2013) in the case of ii) and iii).

The focus then shifts to inference on the underlying parameter vector, with the alternative filters 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 PMCMC algorithm.[1] In so doing we follow the spirit of the exercise undertaken in Pitt *et al.* (2012), 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. Through this exploration of the performance of the various filters, in the PMCMC setting and under a range of different signal-to-noise ratios, important new insights are gained into the relative advantages of competing methods.

The outline of the paper is as follows. In Section 2 we give a brief outline of the role played by particle filtering in likelihood estimation. In Section 3 we then introduce the data-driven filters

[1] To keep the scope of the paper manageable we focus only on filtering-based estimates of the likelihood function as embedded within a so-called marginal MH-based PMCMC algorithm. Other PMCMC approaches, such as particle Gibbs, require particle smoothing techniques, and hence bring a host of other issues to bear on the problem. For recent developments on this problem, see Lindsten, Jordan and Schön (2014) and Chopin and Singh (2015).

that are the focus of the paper, along with theorems which establish that the DPF-based filters yield unbiased estimators of the likelihood function and, hence, serve as an appropriate basis for a PMCMC scheme. The proof of one of the two theorems is given in an appendix to the paper, whilst the proof of the other is shown to follow from Pitt *et al.* (2012) under given conditions. An extensive simulation exercise is conducted in Section 4 based on the three SSMs listed above, and with the performance of the new filters compared with that of filters which feature most prominently in the literature. An empirical study, in which competing PMCMC algorithms are used to produce posterior inference on the parameters of a stochastic volatility model for Australian/US exchange rate returns, is then documented in Section 5. Section 6 concludes.

2 FILTERING-BASED LIKELIHOOD ESTIMATION

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), $p(x_{t+1}|x_t, \theta)$ and with initial state probability given by $p(x_0|\theta)$, 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 $p(y_t|x_t, \theta)$. 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,

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

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, $p(x_t|y_{1:t})$, with those draws in turn being used, via standard calculations, to estimate the prediction densities of which the likelihood function in (1) 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 independent filters 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 true 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

$$E_u[\widehat{p}_u(y_{1:T}|\theta)] = p(y_{1:T}|\theta), \tag{2}$$

i.e., that $\widehat{p}_u(y_{1:T}|\theta)$ is an unbiased estimator of the likelihood function, then the marginal posterior associated with the joint distribution, $p(\theta, u|y_{1:T}) \propto p(y_{1:T}|\theta, u) \times p(\theta) \times p(u)$, 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 replaces the exact (but unavailable) $p(y_{1:T}|\theta)$.

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 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.

3 NEW DATA-DRIVEN FILTERING ALGORITHMS

3.1 OVERVIEW

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

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

$$p(x_{t+1}|x_t, y_{1:t+1}) \propto p\left(y_{t+1}|x_{t+1}\right) p\left(x_{t+1}|x_t\right). \tag{3}$$

Note that draws of the conditioning state value x_t are, at time $t+1$, available from the previous iteration of the filter. To keep notation as simple as possible, explicit dependence of all expressions on the unknown static parameter θ is suppressed until required in Section 3.4.

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

$$g(x_{t+1}|x_t, y_{1:t+1}), \tag{4}$$

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

$$\widehat{p}(x_{t+1}|y_{1:t+1}) = \sum_{j=1}^{N} \pi_{t+1}^{(j)} \delta_{x_{t+1}^{(j)}}, \tag{5}$$

where $\delta_{(\cdot)}$ denotes the (Dirac) delta function, see Au and Tam (1999).[2] 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 IPF-style algorithms. In the case of the BPF the proposal density in (4) is equated to the transition density, $p(x_{t+1}|x_t)$, 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). For the IPF of Lin *et al.* (2005), the proposal reflects the form of $p\left(y_{t+1}|x_{t+1}\right)$ 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 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, $g(x_{t+1}, k|x_t, y_{1:t+1})$, 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 $p\left(y_{t+1}|x_{t+1}\right) p\left(x_{t+1}|x_t\right)$ in (3), the component - either $p\left(y_{t+1}|x_{t+1}\right)$ or $p\left(x_{t+1}|x_t\right)$ - that is relatively more concentrated as a function of the argument x_{t+1}, will dominate

[2] 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).

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 $p\left(y_{t+1}|x_{t+1}\right)$ highly peaked around x_{t+1} as a consequence, an IPF proposal, which attempts to mimic $p\left(y_{t+1}|x_{t+1}\right)$ alone, can yield an accurate estimate of $p(x_{t+1}|x_t, y_{1:t+1})$, 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. Our basic DPF is an IPF and, as such, produces draws of x_{t+1} via the structure of the measurement density alone. The UDPF then augments the information from $p\left(y_{t+1}|x_{t+1}\right)$ with information from the second component in (3).

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,

$$y_{t+1} = h\left(x_{t+1}, \eta_{t+1}\right), \tag{6}$$

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

$$p(y_{t+1}|x_{t+1}) = \int_{-\infty}^{\infty} p(\eta_{t+1}) \left|\frac{\partial h}{\partial x_{t+1}}\right|^{-1}_{x_{t+1}=x_{t+1}(y_{t+1},\eta_{t+1})} \delta_{x_{t+1}(y_{t+1},\eta_{t+1})} d\eta_{t+1}, \tag{7}$$

where $x_{t+1}(y_{t+1}, \eta_{t+1})$ is the unique[3] solution to $y_{t+1} - h(x_{t+1}, \eta_{t+1}) = 0$. Further discussion of the properties of the representation in (7) are provided in Ng *et al.* The advantage of the representation in (7) 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 THE BASIC DATA-DRIVEN PARTICLE FILTER (DPF)

With reference to (6), 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

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

role played by the Jacobian in (7), the particles $x_{t+1}^{(j)}$ serve as a set of independent draws from a proposal distribution with density $g(.)$ satisfying

$$g(x_{t+1}|y_{t+1}) = \left| \frac{\partial h\left(x_{t+1}, \eta_{t+1}\right)}{\partial x_{t+1}} \right|_{\eta_{t+1}=\eta^*(x_{t+1}, y_{t+1})} p\left(y_{t+1}|x_{t+1}\right), \tag{8}$$

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 (8), it is sufficient to assume both partial derivatives of $h\left(\cdot, \cdot\right)$ exist and are non-zero, as occurs in the range of applications considered here. In such settings, and given the lack of explicit dependence of $g(.)$ on x_t, the resultant sample from (8) 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

$$w_{t+1}^{(j)} = \pi_t^{(j)} \frac{p\left(y_{t+1}|x_{t+1}^{(j)}\right) p\left(x_{t+1}^{(j)}|x_t^{(j)}\right)}{g\left(x_{t+1}^{(j)}|y_{t+1}\right)}, \tag{9}$$

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

$$w_{t+1}^{(j)} = \pi_t^{(j)} \left| \frac{\partial h}{\partial x_{t+1}} \right|^{-1}_{\eta_{t+1}=\eta_{t+1}^{(j)},\ x_{t+1}=x_{t+1}^{(j)}} p\left(x_{t+1}^{(j)}|x_t^{(j)}\right), \tag{10}$$

for $j = 1, 2, ..., N$, where $x_0^{(j)} \overset{iid}{\sim} p\left(x_0\right)$, $\pi_0^{(j)} = \frac{1}{N}$, and the filtering weights $\pi_{t+1}^{(j)}$, in (5), are produced sequentially as

$$\pi_{t+1}^{(j)} \propto w_{t+1}^{(j)} \tag{11}$$

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

$$\widehat{p}_u(y_{t+1}|y_{1:t}) = \sum_{j=1}^{N} w_{t+1}^{(j)}, \tag{12}$$

with each $w_{t+1}^{(j)}$ as given in (9).

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 $x_{t+1}^{(j)}$,

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

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

$$w_{t+1}^{(j)} = \frac{1}{L}\left|\frac{\partial h}{\partial x_{t+1}}\right|^{-1}_{\eta_{t+1}=\eta_{t+1}^{(j)},\ x_{t+1}=x_{t+1}^{(j)}} \sum_{l=1}^{L}\left[\pi_t^{(k_{l,j})} p\left(x_{t+1}^{(j)}|x_t^{(k_{l,j})}\right)\right], \tag{13}$$

with $\pi_t^{(j)}$ available from the previous iteration of the filter. Accordingly, as for the $L = 1$ matching case in (11), the $\pi_{t+1}^{(j)}$ are then set proportional to the $w_{t+1}^{(j)}$ in (13), 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 the precision of the resulting likelihood function estimates.[4] To ensure ease of implementation, pseudo code for the DPF algorithm is provided in Algorithm 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.

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

1. Generate $x_0^{(j)}$ from the initial state distribution $p(x_0)$, for $j = 1, 2, ..., N$.
2. Set the normalized particle weight $\pi_0^{(j)} = \frac{1}{N}$.
3. **for** $t = 0, 1, ..., T-1$:
4. Generate $\eta_{t+1}^{(j)} \overset{i.i.d.}{\sim} p(\eta_{t+1})$, for $j = 1, 2, ..., N$.
5. Calculate $w_{t+1}^{(j)}$ according to (13). Note that when $L = 1$ this is equivalent to (10).
6. Calculate $\hat{p}_u(y_{t+1}|y_{1:t})$ using (12).
7. Calculate the normalized particle weight $\pi_{t+1}^{(j)} = \frac{w_{t+1}^{(j)}}{\sum_{i=1}^{N} w_{t+1}^{(j)}}$.
8. Resample N $x_{t+1}^{(j)}$ particles with probabilities $\pi_{t+1}^{(j)}$.
9. Set $\pi_{t+1}^{(j)} = \frac{1}{N}$.

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. 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 (8)

[4] 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).

for the latent state x_{t+1} is obtained without any knowledge of the predictive density (or 'prior') given by $p\left(x_{t+1}|y_{1:t}\right)$, the resampling of the proposed particles according to either (10) or (13), means that the resampled draws themselves *do* take account of this predictive information, and thus appropriately represent the filtered distribution as in (5). Nevertheless, in avoiding the use of past particles when proposing new draws, the performance of the DPF will depend 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.

3.3 THE UNSCENTED DATA-DRIVEN PARTICLE FILTER (UDPF)

Recognizing that the incremental target distribution in (3) may be expressed as

$$p\left(x_{t+1}|x_t, y_{1:t+1}\right) \propto p(y_{t+1}|x_t)p(x_{t+1}|x_t, y_{t+1}), \tag{14}$$

the APF algorithm of Pitt and Shephard (1999) proposes particles by exploiting the form of both components on the right hand side of (14). 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 $p(y_{t+1}|x_t)$ can be computed and $p(x_{t+1}|x_t, y_{t+1})$ is able to be simulated 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.

Our newly proposed UDPF also employs unscented transformations, but for the explicit purpose of producing a Gaussian approximation to $p(y_{t+1}|x_{t+1})$, via the inversion of the measurement equation described above, expressed here as

$$\widehat{p}(y_{t+1}|x_{t+1}) \propto \frac{1}{\widehat{\sigma}_{M,t+1}}\phi\left(\frac{x_{t+1} - \widehat{\mu}_{M,t+1}}{\widehat{\sigma}_{M,t+1}}\right). \tag{15}$$

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. Motivation for the unscented method, including all details of the computation of the moments in this case, is provided in Appendix A. For the sake of simplicity, we assume the state transition density to be Gaussian,

$$p(x_{t+1}|x_t) = \frac{1}{\widehat{\sigma}_{P,t+1}}\phi\left(\frac{x_{t+1} - \widehat{\mu}_{P,t+1}}{\widehat{\sigma}_{P,t+1}}\right), \tag{16}$$

where $\widehat{\mu}_{P,t+1}$ and $\widehat{\sigma}^2_{P,t+1}$ are assumed known, given x_t, and the subscript P references the predictive transition equation to which these moments apply. If needed, a Gaussian approximation (e.g. via a further unscented transformation) of an initial non-Gaussian state equation may be accommodated. Having obtained the Gaussian approximation in (15) and applying the usual conjugacy algebra, a proposal density is constructed as

$$g(x_{t+1}|x_t^{(j)}, y_{t+1}) = \frac{1}{\widehat{\sigma}^{(i)}_{t+1}} \phi\left(\frac{x_{t+1} - \widehat{\mu}^{(j)}_{t+1}}{\widehat{\sigma}^{(i)}_{t+1}}\right), \tag{17}$$

where $\widehat{\mu}^{(j)}_{t+1} = \left\{\widehat{\sigma}^{2(j)}_{P,t+1}\widehat{\mu}_{M,t+1} + \widehat{\sigma}^2_{M,t+1}\widehat{\mu}^{(j)}_{P,t+1}\right\} / \left\{\widehat{\sigma}^{2(j)}_{P,t+1} + \widehat{\sigma}^2_{M,t+1}\right\}$ and $\widehat{\sigma}^{2(i)}_{t+1} = \widehat{\sigma}^2_{M,t+1}\widehat{\sigma}^{2(j)}_{P,t+1} / \left\{\widehat{\sigma}^{2(i)}_{P,t+1} + \widehat{\sigma}^2_{M,t+1}\right\}$ are the requisite moments of the Gaussian proposal, with the subscript (j) used to reflect the dependence on the j^{th} particle $x_t^{(j)}$. A resulting particle draw $x^{(j)}_{t+1}$ from the proposal in (17) is then weighted, as usual, relative to the target, with the particle weight formula given by,

$$w^{(j)}_{t+1} = \pi^{(j)}_t \frac{p\left(y_{t+1}|x^{(j)}_{t+1}\right) p\left(x^{(j)}_{t+1}|x^{(j)}_t\right)}{\frac{1}{\widehat{\sigma}^{(i)}_{t+1}} \phi\left(\frac{x^{(j)}_{t+1} - \widehat{\mu}^{(j)}_{t+1}}{\widehat{\sigma}^{(i)}_{t+1}}\right)}. \tag{18}$$

The corresponding UDPF likelihood estimator is calculated as per (12). Pseudo code for the UDPF is provided in Algorithm 2.

Algorithm 2 *The UDPF*

1. Generate $x_0^{(j)}$ from the initial state distribution $p(x_0)$, for $j = 1, 2, ..., N$.
2. Set the normalized particle weight $\pi_0^{(j)} = \frac{1}{N}$.
3. **for** $t = 0, 1, ..., T-1$:
4. Calculate $\hat{\mu}_{M,t+1}$ and $\hat{\sigma}^2_{M,t+1}$ according to (A.1) and (A.2).
5. Construct the UDPF proposal distribution using (17), for $j = 1, 2, ..., N$.
6. Generate $x^{(j)}_{t+1}$ from Step 5.
7. Calculate the particle weight according to (18).
8. Calculate $\hat{p}_u(y_{t+1}|y_{1:t})$ using (12).
9. Calculate the normalized particle weight $\pi^{(j)}_{t+1} = \frac{w^{(j)}_{t+1}}{\sum_{i=1}^N w^{(j)}_{t+1}}$.
10. Resample N particles with probabilities $\pi^{(j)}_{t+1}$.
11. Set $\pi^{(j)}_{t+1} = \frac{1}{N}$.

3.4 THE UNBIASEDNESS OF THE DATA-DRIVEN FILTERS

As discussed in Section 2, the unbiasedness condition in (2) is required to ensure that a PMCMC scheme yields the correct invariant posterior distribution for θ. To consider the theoretical properties of the new filters proposed in Sections 3.2 and 3.3, we reinstate in this section the explicit

dependence of the joint density of $y_{1:T}$ on θ. 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, and the UPDF. The collective conditions 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 $\int x_{t+1}^k p(y_{t+1}|x_{t+1})dx_{t+1} < \infty$ 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 $1 \leq L \leq N$ matches, satisfies*

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

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

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 $g(y_{t+1}|x_t^{(j)}) = 1$ and with the proposal distribution, $g(x_{t+1}|x_t^{(j)}, y_{t+1})$, formed as per (17). 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. In contrast, since the multiple matching technique is only available for use with the DPF (and not with the APF), the proof in Pitt *et al.* is not adequate to prove Theorem 1. Hence, we provide all details of the proof of Theorem 1, along with those of two lemmas upon which our proof depends, in Appendix B. Sufficient conditions C1 and C2 are required for this proof, as they ensure that the DPF proposal distribution is a proper probability distribution.

4 SIMULATION EXPERIMENTS

In this section the performance of the data-driven filters is investigated in a controlled setting, and compared against that of three of the most commonly applied filters, namely the BPF, APF and UPF. We document both the precision with which the likelihood function is estimated using each filter, at the known true parameters (Section 4.4) and the mixing performance of the associated PMCMC algorithm (Section 4.5). For both exercises, three state space models are entertained: the linear Gaussian (LG) model that is foundational to all state space analysis, and two non-linear, non-Gaussian models that feature in the empirical finance literature, namely the stochastic conditional duration (SCD) and stochastic volatility (SV) specifications referenced in the Introduction. The SCD and SV models in particular contain non-linear and non-Gaussian elements that render the posterior distributions for these models intractible. We have argued that it would be expected that different particle filtering methods will perform well under different signal-to-noise settings, and hence seek to demonstrate this in the simulation output below. The first simulation exercise holds all parameter values fixed at their true values, and thus the impact of the signal-to-noise ratio on the performance of the various filters is able to be assessed directly. However, in the subsequent PMCMC simulation exercise, the signal-to-noise ratio from the true DGP need not be reflected in the proposal distribution for any iteration as the conditioning parameter draw may be very different from that of the true DGP. In this case, particle filtering methods that are robust to the signal-to-noise settings, in that the associated PMCMC algorithm is demonstrated to be relatively efficient, will be preferred.

4.1 SIMULATION DESIGN AND EVALUATION METHODS

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

$$SNR = \sigma_x^2/\sigma_m^2, \tag{19}$$

where σ_x^2 is the unconditional variance of the state variable, which is available analytically in all cases considered. 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 (19) measures the strength of signal relative to the background

noise in the (appropriately transformed) SSM, for given fixed parameter values.[5]

A design scenario is defined by the combination of the model and corresponding model parameter settings that achieve a given value for (19): low, medium 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. (See also Lin *et al.*, 2005.) 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. When assessing PMCMC performance, undertaken in Section 4.5, 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.

For each design scenario considered in the current section, a single time series of length T is produced and, for each of the competing filtering methods, R likelihood estimates - at the true parameter values - are produced using R independent runs of the relevant filter, each based on $N = 1000$ particles at each $t = 1, 2, ..., T$. The (proportionate) likelihood error (LE) from the r^{th} replication, $r = 1, 2, ..., R$, is computed as

$$LE^{(r)} = \frac{\widehat{p}_u^{(r)}(y_{1:T}|\theta_0) - p(y_{1:T}|\theta_0)}{p(y_{1:T}|\theta_0)}, \tag{20}$$

where $\widehat{p}_u^{(r)}(y_{1:T}|\theta_0)$ denotes the r^{th} likelihood estimate from the given filtering algorithm, evaluated at the specified 'true' parameter vector θ_0, and $p(y_{1:T}|\theta_0)$ denotes the corresponding 'exact' likelihood value, computed using the KF in the case of the linear Gaussian model, and the grid-based 'exact' deterministic filter of Ng *et al.* (2013) in the case of the (transformed) non-linear models.[6] Boxplots are used to summarize the distribution of the $LE^{(r)}$ for each filter.

The PMCMC 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

[5] 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 $\log(p(y_t|x_t))$, see Giordani *et al.* (2011).

[6] The grid-based filtering algorithm developed by Ng *et al.* permits a non-parametric estimate of the ordinates of the density function of the distribution of the measurement error over a specified grid. However in the current setting the form of this density parametrically assumed, and hence the grid-based calculation of the likelihood function under this parametric assumption is implemented. Note that although the transformed non-linear models have an additive structure, they remain non-Gaussian.

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 PMCMC algorithm, this will typically slow the mixing in 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 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. 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.

In order to reduce the computational burden, we restrict attention only to the low and high SNR settings for the PMCMC exercise. Then, for each particular filter, the particle marginal MH algorithm of Andrieu *et al.* (2010) is used to produce a Markov chain with $MHit = 110,000$ 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 for any particular design scenario, and for any specific filter, involves running the filter, conditional on the true parameter set, for R_0 initial replications, each time conditional on an arbitrarily chosen number of particles N_s, with the likelihood estimate $\widehat{p}_u^{(r)}(y_{1:T}|\theta_0)$ recorded for each replication $r = 1, 2, ..., R_0$. The sample variance of the likelihood estimator at the true parameter (vector), denoted by $\sigma^2_{N_s}$, is then calculated from the R_0 likelihood estimates, and the optimum number of particles, denoted by N_{opt}, chosen as

$$N_{opt} = N_s \times \frac{\sigma^2_{N_s}}{0.85}. \tag{21}$$

In other words, any initial particle number, N_s, is scaled according to the extent to which the precision that it is expected to yield (as estimated by $\sigma^2_{N_s}$) varies from the value of 0.85 that is sought. We then record the average computational time (over the $MHit$ iterations of the Markov chain) needed to obtain each likelihood estimate at the drawn parameter values using N_{opt} - which we refer to as to as the average likelihood computing time (ALCT) - in addition to 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

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	Medium	High		Low	Medium	High		Low	Medium	High
σ_η	2.24	1.00	0.45	α	0.67	1.43	6.67	ϕ	-6.61	-7.94	-4.24
ρ	0.40	0.40	0.40	β	1.50	0.70	0.15	ρ	0.2	0.2	0.6
σ_v	0.92	0.92	0.92	ϕ	-1.1	-1.1	-1.1	σ_v	0.70	1.50	1.40
				ρ	0.74	0.74	0.74				
				σ_v	0.65	0.65	0.65				
SNR	0.2	1	5		0.5	1.6	10		0.1	0.47	0.6

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.

4.2 MODELS, SNR SETTINGS 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

$$y_t = x_t + \sigma_\eta \eta_t \tag{22}$$

$$x_t = \rho x_{t-1} + \sigma_v v_t, \tag{23}$$

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 a range of values for SNR (low, medium and high), as recorded in Panel A of Table 1. These parameter settings are then taken as fixed and known for the purpose of evaluating the performance of the likelihood estimators, with the results provided in Section 4.4. In each of the subsequent PMCMC exercises detailed in Section 4.5, 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

$$y_t = \exp(x_t)\eta_t \tag{24}$$

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

with $v_t \sim i.i.d.N(0,1)$ independent of η_t, and with η_t being *i.i.d. gamma* with shape parameter α and rate parameter β. Taking the logarithms of both sides of (24) 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 $x_0 \sim N\left(\frac{\phi}{1-\rho}, \frac{\sigma_v^2}{(1-\rho)^2}\right)$. Again, the original parameter settings are then taken as fixed and known for the purpose of evaluating the likelihood estimators, discussed in Section 4.4, while for the corresponding PMCMC exercise, detailed in Section 4.5, 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 PMCMC performance.

4.2.3 The stochastic volatility (SV) model

The SV model is given by

$$y_t = \exp(x_t/2)\eta_t \tag{26}$$

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

with η_t and v_t once again mutually independent *i.i.d.* standard normal random variables. Once again, to fix the SNR, the measurement equation is transformed to obtain $\log(y_t^2) = x_t + \varepsilon_t$, where $\varepsilon_t = \log(\eta_t^2)$. In this case, $var(\varepsilon_t) = 4.93$, corresponding to the quantity σ_m^2 in (19). As recorded in Panel C of Table 1, three distinct parameter settings are considered for the likelihood estimation exercise, with the SNR for each setting calculated according to (19), and chosen in such a way that the simulated data under each scenario is empirically plausible. The initial state distribution is once again specified as $x_0 \sim N(\frac{\phi}{1-\rho}, \frac{\sigma_v^2}{(1-\rho)^2})$. For the PMCMC 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 and 3.3, respectively. 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 report results for an alternative version of APF (in which full adaptation is not exploited) in which the proposal distribution is given by $g\left(x_{t+1}, k|x_t, y_{1:t+1}\right) = p(y_{t+1}|\mu(x_t^{(k)}))p(x_{t+1}|x_t^{(k)})$, where $\mu(x_t^{(k)})$ is the conditional mean $E(x_{t+1}|x_t^{(k)})$, and k is a discrete auxiliary variable (see Pitt and Shephard, 1999, for details). For the SV model, we also report the results for a third version of APF based on a second order Taylor's series expansion of $\log(p(y_{t+1}|x_{t+1}))$ around the maximum of the measurement density (referred to as TAPF hereafter). This yields an approximation of the likelihood component, denoted by $g(y_{t+1}|x_{t+1})$, which is then used to form a proposal distribution, $g\left(x_{t+1}, k|x_t, y_{1:t+1}\right) = g(y_{t+1}|x_{t+1}, \mu(x_t^{(k)}))p(x_{t+1}|x_t^{(k)})$. (For more details, see Pitt and Shephard, 1999, and Smith and Santos, 2006.) Although conceptually these non-fully-adapted APF methods apply for any model, as noted in Section 4.4 below they do not always result in empirically stable likelihood estimates.

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 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 (4) 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 $p\left(y_{t+1}|x_{t+1}\right)$ and $p\left(x_{t+1}|x_t\right)$. Further discussion of the UPF is provided in van de Merwe *et al.* (2000).

4.4 ACCURACY OF LIKELIHOOD ESTIMATION: RESULTS

The results of the first simulation exercise are summarized in Figures 1, 2 and 3 using boxplots of the replicated likelihood estimation errors in (20) for all filters, under each SNR setting and

for each of the three models considered. A sample size of $T = 50$ is used for the purpose of this exercise, with the number of replications set as $R = 1000$. The bold line in a boxplot represents the median of the relevant likelihood errors while the lower and upper edges of the box represent the first and third quantiles of these errors, respectively. In short, a boxplot that is concentrated around zero and has relatively few outliers, indicates that the relevant filtering method produces a precise estimate of the likelihood value. The ALCT (measured in seconds) required to produce a single likelihood estimate is also reported (in parentheses) under the label of the corresponding filter in the boxplot.

As is evident from Figure 1, and as anticipated, as the SNR increases the likelihood error associated with the DPF becomes more concentrated around zero, indicating that the DPF performs better when the system signal is strong. In contrast, the BPF tends to performs better when the system signal is weak. The ALCT of the BPF (0.02 seconds) is equivalent to that of the DPF with a single ($L = 1$) match, so there is nothing to choose between the filters on that score, for this simple model at least. The variance of the DPF likelihood errors is very similar for $L = 1$ and 30 matches. This indicates that increasing the number of matches (and, hence, the computation time) does not noticeably improve the performance of the DPF for this example. Broadly similar comments apply to the relative performance of the BPF and the DPF in the SCD and SV models, as documented in Figures 2 and 3 respectively, except that the ALCT for the DPF with a single match is uniformly smaller (0.02 seconds) than that of the BPF (0.03 seconds) for both non-linear models. [7]

The UDPF, on the other hand, again as anticipated, demonstrates a performance that is very robust to the SNR value, and which (for all three models) is superior to that of all alternative filters other than the 'gold-standard' fully-adapted APF (FAPF) in the LG setting. Indeed, with the exception of the latter, all filters other than the UDPF are seen to produce either a large variance in the likelihood error, or an excessive proportion of outliers, or both, in at least one of the SNR (and model) settings considered. In particular, it can be seen that in all three models accuracy of the APF likelihood estimate (plus the TAPF likelihood estimate documented for the SV model) declines as the strength of the signal increases. The decline in accuracy is manifested in terms of one or more of: an increase in variance, a median value that is further from zero, or an increase in the frequency of extreme errors (outliers). All forms of inaccuracy reflect the fact that the APF/TAPF proposal matches the target distribution more poorly as the signal gets stronger. The extent of the inaccuracy for the TAPF in the SCD model is such that the associated results

[7] For other evidence on this point, in different model settings, see Lin *et al.* (2005) and Klaas *et al.* (2012).

are not recorded at all in Figure 2.

4.5 PMCMC PERFORMANCE: RESULTS

At each MCMC iteration, the particle filter as 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, all other things equal, tend to produce a small value for the ALCT and, hence, 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 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 a reasonable sample size when comparing the performance of competing *inferential* algorithms in a state space setting.[8]

The PMCMC results for the LG model are presented in Table 2. As is consistent with the likelihood estimation results documented in the previous section, 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

[8] Given the overall inaccuracy documented for the APF and TAPF it was anticipated that N_{opt} and, hence, the ALCT, for these filters would be too large to justify their inclusion in the comparative PMCMC exercise; they have therefore been omitted in both this section and the following empirical section.

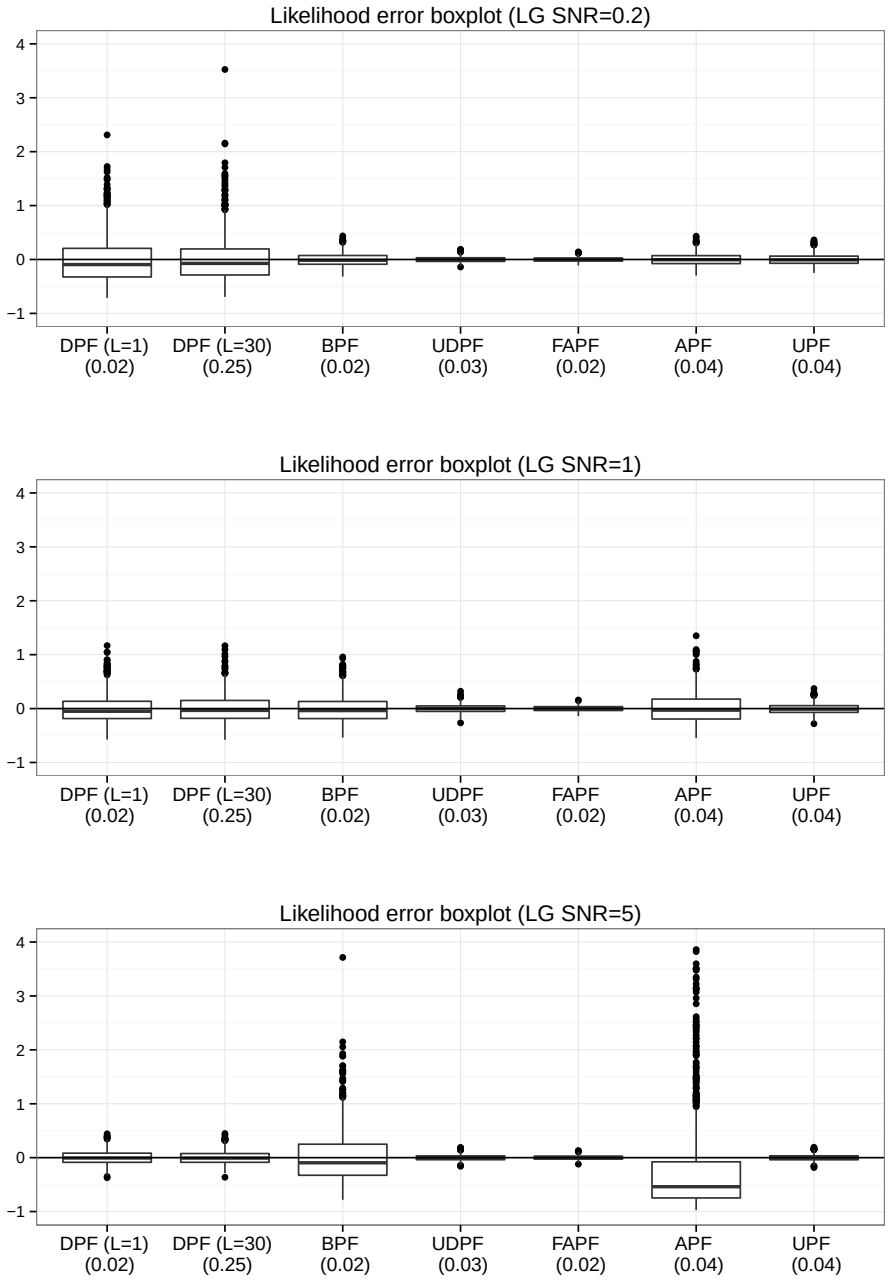

Figure 1: Likelihood error boxplot of DPF (with $L = 1$ and 30 matches), BPF, UDPF, FAPF, APF and UPF based on $R = 1000$ replications. The particle size is $N = 1000$ for all filters. Data is simulated from the linear Gaussian (LG) model with low SNR (top panel), medium SNR (middle panel) and high SNR (bottom panel). The average likelihood computational time (ALCT) of each filter is reported in parentheses under the label of the filter.

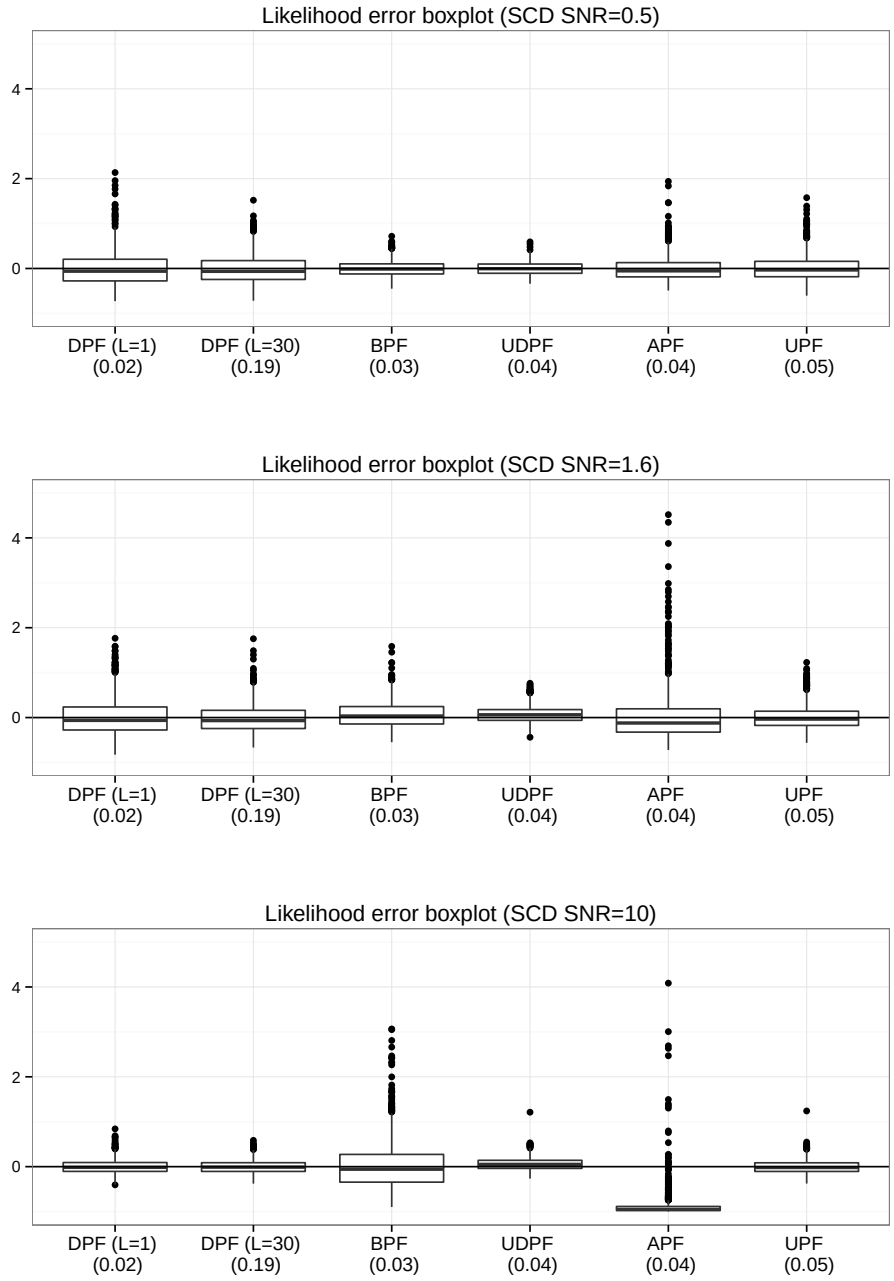

Figure 2: Likelihood error boxplot of DPF (with $L = 1$ and 30 matches), BPF, UDPF, APF and UPF based on $R = 1000$ replications. The particle size is $N = 1000$ for all filters. Data is simulated from the stochastic conditional duration (SCD) model with low SNR (top panel), medium SNR (middle panel) and high SNR (bottom panel). The average likelihood computational time (ALCT) of each filter is reported in parentheses under the label of the filter.

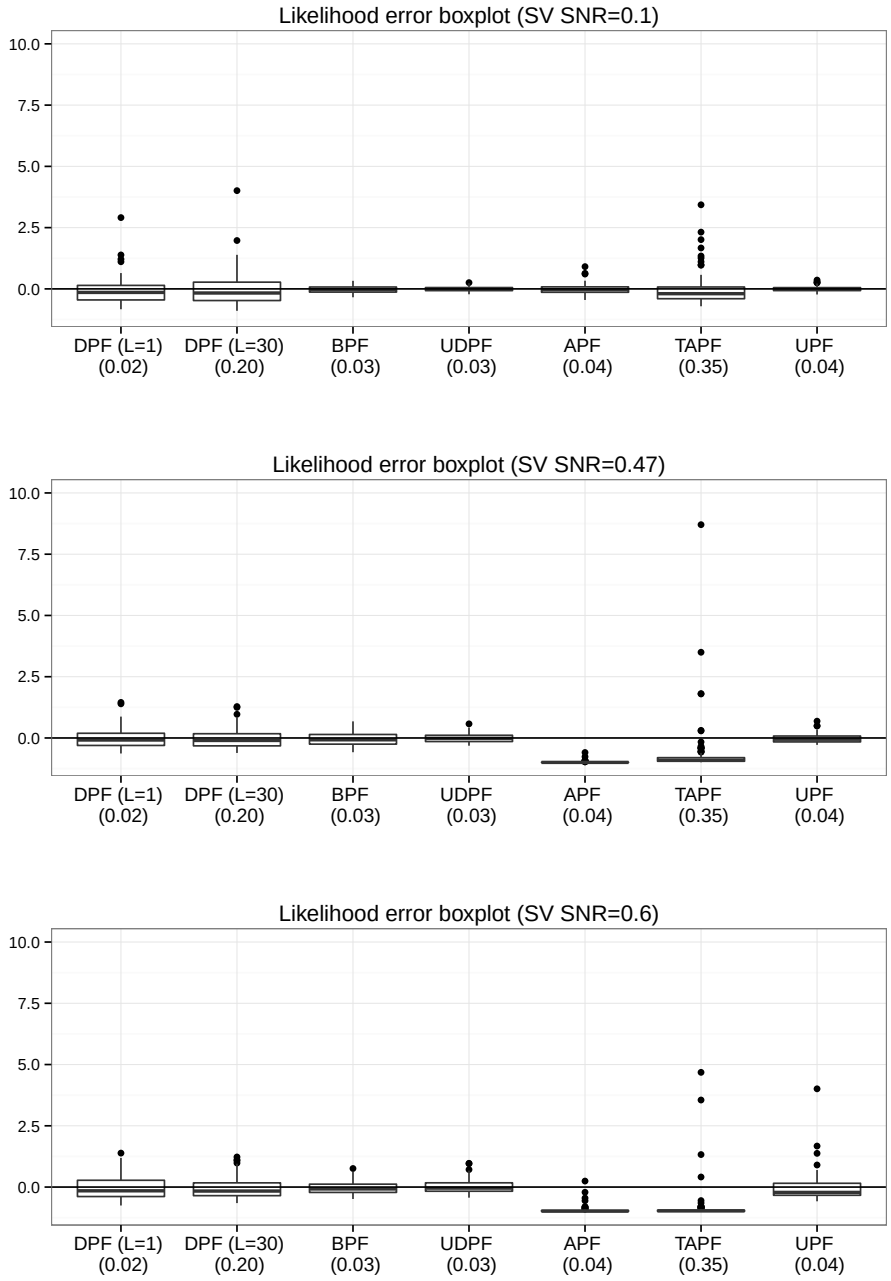

Figure 3: Likelihood error boxplot of DPF (with $L = 1$ and 30 matches), BPF, UDPF, APF, TAPF and UPF based on $R = 1000$ replications. The particle size is $N = 1000$ for all filters. Data is simulated from the stochastic volatility (SV) model with low SNR (top panel), medium SNR (middle panel) and high SNR (bottom panel). The average likelihood computational time (ALCT) of each filter is reported in parentheses under the label of the filter.

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, UDPF, FAPF and UPF 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.

PMCMC 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
UDPF	4	0.065	134.2	63.3	149.9
FAPF	2	0.022	163.8	98.4	181.6
UPF	57	0.057	128.2	88.3	144.5

PMCMC 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
UDPF	23	0.060	37.4	35.5	39.9
FAPF	11	0.035	35.0	35.4	39.8
UPF	70	0.066	31.6	31.6	32.2

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, UDPF and UPF 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.

PMCMC 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	2.1	47.7	45.6	38.8	37.2	45.9
BPF	184	0.064	58.3	57.3	46.1	40.0	42.6
UDPF	119	0.065	45.9	55.9	52.3	36.8	55.1
UPF	398	0.239	55.2	67.1	49.6	42.3	66.3

PMCMC 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, UDPF and UPF 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.

PMCMC results under a low SNR setting

	N_{opt}	ALCT	IF		
			ϕ	ρ	σ_v^2
DPF (L=1)	1767	0.534	22.53	22.42	19.07
DPF (L=30)	1548	3.37	21.39	21.38	18.59
BPF	275	0.067	14.82	14.83	16.17
UDPF	245	0.085	14.55	14.62	13.97
UPF	244	0.133	22.35	22.32	21.35

PMCMC results under a high SNR setting

	N_{opt}	ALCT	IF		
			ϕ	ρ	σ_v^2
DPF (L=1)	1013	0.265	14.85	15.39	15.45
DPF (L=30)	915	1.889	17.59	18.25	16.39
BPF	605	0.104	16.86	16.62	14.94
UDPF	568	0.156	12.55	12.91	13.81
UPF	686	0.256	16.41	16.53	18.84

the unscented transformations) resulting in only a slightly larger value for ALCT (relative to that for the BPF) in the low SNR case. Moreover, the UDPF yields very similar values of N_{opt} to the fully analytical 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 likelihood estimation (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 PMCMC 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 documented 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.

4.6 Discussion

Thus far we have only considered the application of the various filters alone, and when the models applied were correctly specified. However, we note that in empirical settings it may be useful to combine our DPF and/or UDPF with other techniques. For example, as noted by Fox *et al.* (2001), draws from a mixture of different proposal distributions will typically diversify the particle set and, hence, result in a lower approximation error than if a single particle filtering methodology is employed. Indeed, other existing strategies could also be useful for diversifying the collection of active particles, such as incorporating MCMC moves (e.g. Gilks and Berzuini, 1999)

or 'look-ahead' strategies (e.g. Del Moral and Murray, 2015). We would anticipate that there will be situations, particularly when the measurement noise is relatively small, when the data-driven methods proposed here will provide some gains.

We also emphasize the value in exploring the performance of various filtering or other advanced MCMC strategies proposed for state space models under a range of different SNR simulation settings. While not always sufficient to explain all variation in performance, we do feel that is offers some insights into the relative advantages of competing methods.

5 EMPIRICAL ILLUSTRATION

Motivated by the performance of the UDPF in relevant non-linear settings, we complete the paper with an empirical application in which we use this particular filter within a PMCMC algorithm to estimate the SV model in (26) and (27), using weekly AUD/USD exchange rate return data from January 14, 2003 to August 4, 2015. A time series plot of the $T = 644$ observations (Figure 4) over this period indicates the usual volatility clustering that justifies the specification of a time-varying volatility model, with an apparent lack of extreme tail behavior motivating our simple choice of conditional Gaussianity for the measurement noise. The BPF is included as a comparator as it represents the simplest filter that may be adopted in this setting, and the UPF included as the filter that is (overall) the most competitive with the UDPF in the simulation experiments. Due to the computational problems flagged earlier, no version of the APF is applied in this exercise. As per the description in Section 4.1 we choose N_{opt} via preliminary runs of $R = 100$ replications of each filter. In this empirical setting, however, in which the true parameter values are unknown, we perform this computation at the posterior mean estimated based on a preliminary PMCMC exercise, in which $N = 1000$ particles are used to estimate the likelihood function. This number of particles is held fixed throughout the subsequent PMCMC algorithm. As in the simulation experiments, we report the resulting ALCT and the inefficiency factors for each method.

The results recorded in Table 5 demonstrate that, in comparison with the UPF, the UDPF requires fewer particles, and less computational time, to produce a chain with notably greater efficiency. Consistent with the simulation results, the SV setting - in which the SNR is likely to be on the low side - favours the BPF in terms of ALCT. However, the UDPF still requires a smaller number of particles to create a chain with efficiency that is comparable to that produced by the BPF.

Table 5: The optimal number of particles, average likelihood computing time (ALCT) and inefficiency factor (IF) for PMCMC algorithms applied to a SV model for weekly AUD/USD exchange rate returns: 14-Jan-2003 to 4-Aug-2015. The BPF, UDPF and UPF, respectively, is used to produce the likelihood estimate.

	PMCMC results				
	N_{opt}	ALCT	IF		
			ϕ	ρ	σ_v^2
BPF	856	0.447	18.97	18.92	21.45
UDPF	584	0.692	26.70	26.72	32.76
UPF	2241	1.654	63.34	63.74	73.87

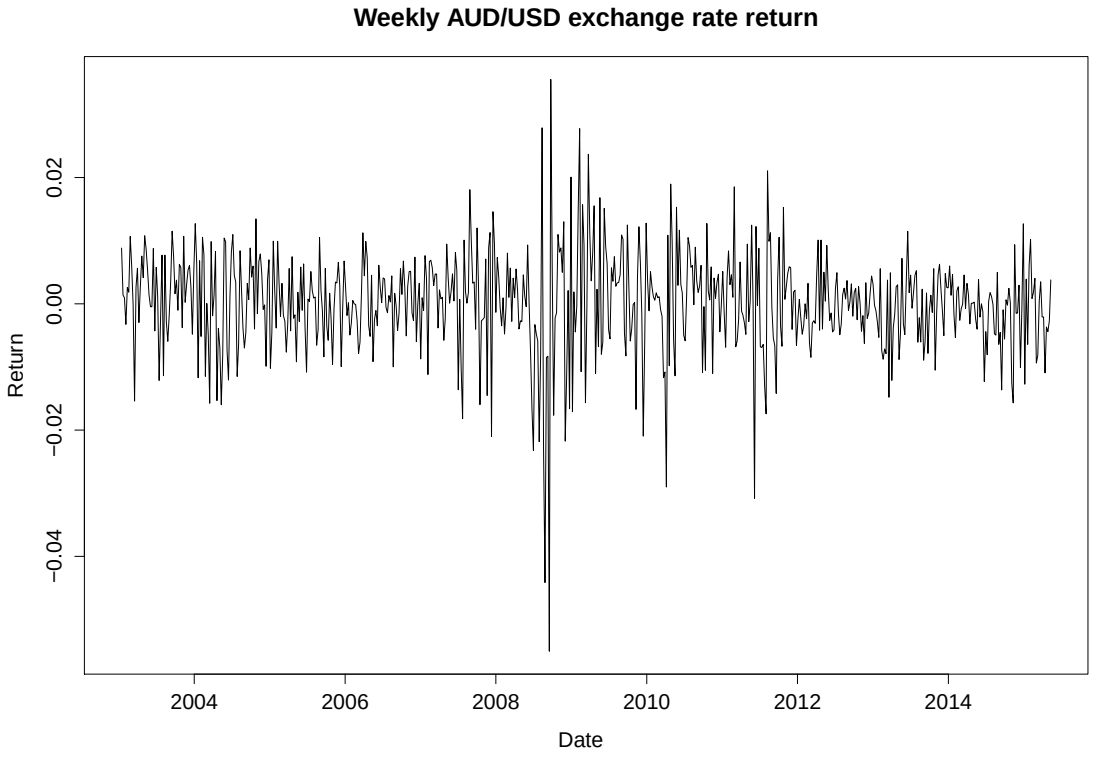

Figure 4: Time series plot of the 644 weekly AUD/USD exchange rate return data from 14-Jan-2003 to 4-Aug-2015.

6 CONCLUDING REMARKS

Inference regarding the parameters of a non-linear or non-Gaussian state space model is challenging. This paper proposes two new particle filtering algorithms that exploit a particular representation of the measurement equation, and compare their performance in a variety of settings with several filters that currently feature in the literature. Given the spirit of their derivation we refer to both new filters as 'data-driven'. Using simulation, we show that the basic form of the data-driven filter (DPF) performs well relative to comparators under a strong SNR scenario, but that its relative performance is adversely affected (as anticipated) by a low SNR. Whilst the technique of multiple matching can, in principle, improve the accuracy of likelihood estimation based on the DPF, no evidence of such an improvement was discerned for the particular models explored herein, with the extra computational burden thus not appearing to reap benefits. The unscented DPF (UDPF) employs the idea of adaptation and is shown to be accurate, as a basis for likelihood estimation, under a wide range of SNR and model settings, and in comparison with a range of existing filters that

includes various versions of the APF. We proceed to demonstrate - both via simulation and through an empirical illustration - that the UDPF, when used within a PMCMC algorithm, continues to perform well relative to other filters, when robustness of performance to SNR, computational time and PMCMC mixing are all taken into account. This augers well for the use of this particular filter in state space settings beyond those explored herein.

A APPENDICES

A.1 THE USE OF 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 *et al.*, 1995, 2000.)

In the UDPF, the unscented transformation is applied to the function defined by solving the measurement equation in (6) 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 (15), 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

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

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

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

where $C_{t+1} = (\int p(y_{t+1}|x_{t+1})dx_{t+1})^{-1}$ represents the normalizing constant that ensures a proper density. Further, using the Dirac representation of the measurement density in (7), we have

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

Using then the discrete approximation of $p(\eta_{t+1})$ 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

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

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

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

A.2 THE PROOF OF THE UNBIASEDNESS OF THE DPF LIKELIHOOD ESTIMATOR (THEOREM 1)

We adapt the proof from Pitt *et al.* (2012) in order 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 (12) for each $t = 1, 2, ..., T$, with the weights $w^{(j)}_{t+1}$ defined by the relevant particle filtering algorithm.

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 in Section 3.4, the conditions outlined for this theorem ensure that $g(x_t|y_t)$ 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 (5), 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

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

Lemma 2

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

According to Section 3.2, the estimator of the likelihood component for the DPF (with potential multiple matching), for given $1 \leq L \leq N$, is

$$\begin{aligned}
\widehat{p}_u(y_t|y_{1:t-1}) &= \sum_{j=1}^{N} w_t^{(j)} \\
&= \sum_{j=1}^{N} \left(\frac{1}{L} \sum_{l=1}^{L} w_t^{(j)(l)} \right) \\
&= \sum_{j=1}^{N} \left(\frac{1}{L} \sum_{l=1}^{L} \frac{p(y_t|x_t^{(j)}) \pi_{t-1}^{(k_{l,j})} p(x_t^{(j)}|x_{t-1}^{(k_{l,j})})}{g(x_t^{(j)}|y_t)} \right),
\end{aligned} \tag{A.5}$$

where the proposal distribution is given in (8) and $k_{l,j}$ represents the j^{th} component of the l^{th} cyclic permutation, K_l, as defined in Section 3.2.

Proof of Lemma 1. We start with,

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

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:

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

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

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

Hence, Lemma 1 holds. ■

Proof of Lemma 2. To prove Lemma 2, we use method of induction as per Pitt *et al.* 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

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

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

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

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

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

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

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

as required. ■

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

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

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

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

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

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

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

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] Au, C. and Tam, J. 1999. Transforming Variables Using the Dirac Generalized Function. *The American Statistician* 53, 270-272.

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Gilks, W.R. and Berzuini, C. 2001. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63(1), pp. 127-146.

[13] 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.

[14] 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.

[15] Guarniero, P., Johansen, A.M. and Lee, A. 2016. The iterated auxiliary particle filter. *arXiv preprint arXiv:1511.06286.*

[16] Hannig, J. Iyer, H., Lai, R.C.S. and Lee, T.C.M. (2016). Generalized Fiducial Inference: A Review and New Results. To appear in *Journal of the American Statistical Association*, doi: 10.1080/01621459.2016.1165102, 56 pages.

[17] Julier, S.J. and Uhlmann, J.K. 1997. New extension of the Kalman filter to nonlinear systems. *AeroSense'97* , pp. 182-193.

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

[19] 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.

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

[21] 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.

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

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

[24] 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

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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