Sequential Bayesian Inference for Dynamic State Space Model Parameters

Arnab Bhattacharya and Simon Wilson*

Abstract

A method for sequential Bayesian inference of the static parameters of a dynamic state space model is proposed. The method uses filtering and prediction distribution approximations such as the extended and unscented Kalman filters; any other valid approximation can also be used. The method computes the posterior distribution on a discrete grid that tracks the support dynamically. Simulation studies show that the method provides a good trade off between computation speed and accuracy, relative to the integrated nested Laplace approximation and a particle filter, in examples of both non-linear and non-Gaussian models.

Keywords: dynamic state space models, sequential estimation, Bayesian inference, grid-based methods, static parameter estimation

*Arnab Bhattacharya is Post Doctoral researcher, School of Computer Science and Statistics, Lloyd Institute, Trinity College, Dublin 2, Ireland (e-mail: bhattaa@tcd.ie); and Simon P. Wilson is Professor, School of Computer Science and Statistics, Lloyd Institute, Trinity College, Dublin 2, Ireland (e-mail: simon.wilson@tcd.ie). This work is supported by the STATICA project, funded by the Principal Investigator programme of Science Foundation Ireland, contract number 08/IN.1/I1879.
1. INTRODUCTION

Dynamic state-space models (West and Harrison [1997]), consisting of a latent Markov process $X_0, X_1, \ldots$ and noisy observations of that process $Y_1, Y_2, \ldots$ that are conditionally independent, are used in a wide variety of applications e.g. wireless networks (Haykin et al., 2004), object tracking (Ristic et al., 2004), econometrics (Hamilton, 1986) etc. The model is specified by an initial distribution $p(x_0|\theta)$, a transition kernel $p(x_t|x_{t-1}, \theta)$ and an observation distribution $p(y_t|x_t, \theta)$. These distributions are defined in terms of a set of $K$ static (e.g. non-time varying) parameters $\theta = (\theta_1, \ldots, \theta_K)$. The joint model to time $T$ is:

$$p(y_{1:T}, x_{0:T}, \theta) = \left( \prod_{t=1}^{T} p(y_t|x_t, \theta)p(x_t|x_{t-1}, \theta) \right) \times p(x_0|\theta)p(\theta), \quad (1.1)$$

where $y_{1:T} = (y_1, \ldots, y_T)$, etc. These models are also known as hidden Markov models (Rabiner, 1989).

In this paper we focus on sequential Bayesian estimation of $\theta$ for these models; at time $T$, we observe $y_T$ and wish to update our inference about $\theta$ from $p(\theta|y_{1:T-1})$ to $p(\theta|y_{1:T})$. Further, this is to be done in a setting where real-time or otherwise fast estimation is required, so that there is an issue of trade-off between computation speed and accuracy. The constraint on computation time means that for some sufficiently large $T$ it becomes infeasible to simply recompute $p(\theta|y_{1:T})$ “from scratch” as proportional
to \( \int p(y_{1:T}, x_{0:T}, \theta)dx_{0:T} \) by Monte Carlo (e.g. MCMC) or a functional approximation (e.g. the integrated nested Laplace approximation) or even by some sequential Monte Carlo method like a particle filter (Gordon et al., 1993) adapted to also infer these static parameters (Ionides et al., 2011; Andrieu et al., 2010). Additionally for many real-time applications, there may be constraints on storing the entire data sequence \( y_{1:T} \) as \( T \) grows large. We propose a method that accomplishes this for a fairly broad set of dynamic state space models.

This focus is in contrast to most of the literature on inference for these models, which concentrate on computation of predictive distributions \( p(x_T | y_{1:T-1}, \theta) \) and \( p(y_T | y_{1:T-1}, \theta) \), and the filtering distributions \( p(x_T | y_{1:T}, \theta) \). Updating of these distributions is by the well-known ‘forward’ equations

\[
p(x_T | y_{1:T-1}, \theta) = \int p(x_T | x_{T-1}, \theta)p(x_{T-1} | y_{1:T-1}, \theta)dx_{T-1}, \quad (1.2)
\]

\[
p(y_T | y_{1:T-1}, \theta) = \int p(y_T | x_T, \theta)p(x_T | y_{1:T-1}, \theta)dx_T \quad \text{and} \quad (1.3)
\]

\[
p(x_T | y_{1:T}, \theta) = p(y_T | x_T, \theta) \times \nonumber \frac{p(x_T | y_{1:T-1}, \theta)}{\int p(y_T | x_T, \theta)p(x_T | y_{1:T-1}, \theta) \, dx_T}. \quad (1.4)
\]

These inferences are usually done conditional on \( \theta \), which is either assumed known or derived as a point estimate of an off-line method. Sequential Monte Carlo methods for these models also concentrate on the filtering distributions conditional on \( \theta \).

The Kalman filter (Kalman, 1960) is one of the earliest and best known dynamic state-space models. The Gaussian form of the model allows filtering and prediction to
be done quickly, exactly and sequentially, without the need to store the data sequence (Meinhold and Singpurwalla, 1983). Extending inference to non-linear and/or non-Gaussian models has proved to be challenging, and there are many approaches in the literature. Examples of functional approximation approaches are the extended Kalman filter (Haykin, 2001) and the unscented Kalman filter (Julier and Uhlmann, 1997). Monte Carlo approaches are mainly sequential e.g. the particle filter (Doucet et al., 2001) and more recently MCMC (Andrieu et al., 2010).

As regards the static parameter estimation problem, almost no closed form solutions are available, even in linear Gaussian models (Andrieu et al., 2005). West and Harrison (1997) show that conjugate sequential updates of the state and observation variances, as well as for $x_0$, are available for some specific cases. To allow for static parameter estimation within sequential Monte Carlo methods, $\theta$ is usually considered to be in fact time-varying (Poyiadjis et al., 2011). Kantas et al. (2009) is a good overview of parameter estimation, including both off-line approaches such as maximum likelihood and the use of sequential Monte Carlo for on-line estimation.

The paper is organized as follows. Section 2 outlines the principle of the method. Sections 3 and 4 describe the 2 main issues to be resolved in order to implement the method: approximations to one-step ahead filtering and prediction densities, and updating of the grid on which the posterior is computed. Section 5 illustrates the method and assesses its performance against alternative approaches. Section 6 contains some concluding remarks.
2. PRINCIPLE

The principle of the proposed method is based on two observations.

The first observation is that many dynamic state space models have a relatively small number of static parameters, so that in principle \( p(\theta|y_{1:T}) \) can be computed and stored on a discrete grid of practical size. This has been noted as a property of many latent models Rue et al. (2009). It is noted that the transition kernel of some dynamic state space models is itself defined in terms of \( \theta \) and time-varying parameters \( \psi_t \) that also evolve as a Markov process e.g. the transition kernel is \( p(x_t|x_{t-1}, \psi_t, \theta) \) and \( p(\psi_{0:T}|\theta) = p(\psi_0|\theta) \prod_{t=1}^T p(\psi_t|\psi_{t-1}, \theta) \); for example dynamic linear models with a trend (West and Harrison 1997). Without loss of generality, the observation is also applicable to this case by considering \((x_t, \psi_t)\) to be the latent process.

The second observation is that there are useful identities for parameter estimation in latent models. One is

\[
p(\theta|y_{1:T}) \propto p(y_{1:T}, \theta) = p(y_{1:T}, x_{0:T}, \theta)/p(x_{0:T}|y_{1:T}, \theta)|_{x_{0:T}=x^*(\theta)},
\]

valid for any \( x_{0:T} \) for which \( p(x_{0:T}|y_{1:T}, \theta) > 0 \). This is the basis of the integrated nested Laplace approximation (INLA) of Rue et al. 2009, where a Gaussian approximation is made for the denominator term, and it is evaluated on a discrete grid of values of \( \theta \); the method also includes a way to derive such a grid. The value \( x_{0:T} = x^*(\theta) \) is
allowed to be a function of $\theta$ and typically $x^*(\theta) = \text{arg max}_{x_{0:T}} p(x_{0:T}|y_{1:T}, \theta)$ is used; in INLA this is just the mean of the Gaussian approximation of $p(x_{0:T}|y_{1:T}, \theta)$. This has been shown to be very accurate when $p(x_{0:T}|\theta)$ is Gaussian but it has an obvious disadvantage in the sequential estimation setting that the dimension of the distribution to be approximated grows with $T$.

Another useful identity is:

$$p(\theta|y_{1:T}) \propto p(\theta|y_{1:T-1}) p(y_T|x_T, \theta)$$

$$\times p(x_T|y_{1:T-1}, \theta)/p(x_T|y_{1:T}, \theta)|_{x_T=x^*(\theta)}; \quad (2.2)$$

as with Equation 2.1, we choose $x^*(\theta) = \text{arg max}_{x_T} p(x_T|y_{1:T}, \theta)$. This identity is clearly useful for sequential estimation and does not suffer from the dimension-increasing problem of Equation 2.1. Taking Equation 2.2, then if prediction and filtering approximations $\tilde{p}(x_T|y_{1:T-1}, \theta)$ and $\tilde{p}(x_T|y_{1:T}, \theta)$ are available, any approximation $\tilde{p}(\theta|y_{1:T-1})$ at time $T-1$ can be updated:

$$\tilde{p}(\theta|y_{1:T}) \propto \tilde{p}(\theta|y_{1:T-1}) p(y_T|x_T, \theta)$$

$$\times \tilde{p}(x_T|y_{1:T-1}, \theta)/\tilde{p}(x_T|y_{1:T}, \theta)|_{x_T=x^*(\theta)}; \quad (2.3)$$

where $x^*(\theta) = \text{arg max}_{x_T} \tilde{p}(x_T|y_{1:T}, \theta)$. For $\theta$ of low dimension, computing Equation 2.3 on a discrete grid offers the potential for fast sequential estimation.

This suggests the following sequential estimation algorithm when approximate pre-
diction and filtering distributions are available. Initially, \( p(\theta | y_{1:T}) \) is approximated by INLA because it is accurate and produces a discrete grid \( \Theta_T \) over which \( p(\theta | y_{1:T}) \) is computed. At some time \( T_{\text{INLA}} \) this will prove to be too slow to compute, and from then on the sequential update of Equation 2.3 will be used.

Two issues remain to be addressed in order to implement this algorithm: the form of the approximations \( \tilde{p}(x_T | y_{1:T-1}, \theta) \) and \( \tilde{p}(x_T | y_{1:T}, \theta) \) and how to adapt the grid \( \Theta_T \) for \( T \geq T_{\text{INLA}} \). These issues are addressed in the next sections.

### 3. Predicting and Filtering Density Approximations

For the Kalman filter (where \( p(y_t | x_t, \theta) \), \( p(x_0 | \theta) \) and \( p(x_t | x_{t-1}, \theta) \) are linear and Gaussian), the prediction and filtering distributions are Gaussian, Equation 2.2 can be computed exactly, and the INLA approximation is also exact. The means and variances of these Gaussians are sequentially updated (Meinhold and Singpurwalla, 1983). All that we need to store are the means and variances of the prediction and filtering distributions for each \( \theta \) in the grid; from this \( p(\theta | y_{1:T}) \) can be computed.

An equivalent definition of Eq. 1.1, and one that is useful in describing some aspects of the approximations that we propose, is the state-space representation:

\[
y_t = f(x_t, u_t, v_t, \theta); \quad (3.1)
\]
\[
x_t = g(x_{t-1}, w_t, \theta), \quad (3.2)
\]
where \( v_t \) and \( w_t \) are observation and system errors, and \( u_t \) are (possibly non-existent) exogenous variables. The likelihood \( p(y_t|x_t, \theta) \) is specified by \( f \) and \( v_t \), while the transition density \( p(x_t|x_{t-1}, \theta) \) is specified by \( g \) and \( w_t \).

### 3.1 Basic Approximations

When either the linear or Gaussian property does not hold, 2 extensions of the Kalman filter can be computed quickly.

**Extended KF**

The extended Kalman filter was one of the first generalisations of the Kalman filter to non-linear models (McElhoe, 1966). It linearizes a non-linear model to create a Kalman filter (e.g. Gaussian) approximation to the filtering and prediction densities (Haykin, 2001). Hence the prediction and filtering distribution approximations are Gaussian and make use of the fast sequential updating of their mean and variances.

**Unscented KF**

The unscented Kalman filter also produces Gaussian approximations to the filtering and prediction densities but approximately propagates the means and covariances through the non-linear function (Julier and Uhlmann, 1997). It tends to be more accurate than the extended Kalman filter, more so for strongly non-linear models.

The non-linearity in the model is propagated deterministically through a small set of points, known as sigma points. Weights are associated with each point and an
estimate of the mean and variance of the Gaussian approximations to \( p(x_t|y_{1:t-1}, \theta) \) and \( p(x_t|y_t, \theta) \) is made as weighted means and variances of these points; see the Appendix for details. The method is computationally fast as it only requires the updating of these points, and then the means and variances of the approximation, at each observation.

### 3.2 Correction Factors

(Cseke and Heskes 2010) suggest several corrections to a Gaussian approximation of \( p(x_T|y_{1:T}, \theta) \) when it has been derived as the marginal of a joint Gaussian approximation \( \tilde{p}(x_{1:T}|y_{1:T}, \theta) \). The simplest of their corrections is a marginal one of the form

\[
\tilde{p}_c(x_T|y_{1:T}, \theta) = \epsilon_T(x_T, y_T, \theta)\tilde{p}(x_T|y_{1:T}, \theta),
\]

(3.3)

They suggest that, in the case where \( p(x_{1:T}|\theta) \) is Gaussian and \( \tilde{p}(x_{1:T}|y_{1:T}, \theta) \) is derived from making a Gaussian approximation to the likelihood terms:

\[
\tilde{p}(x_{1:T}|y_{1:T}\theta) \propto p(x_{1:T}|\theta) \prod_{t=1}^{T} \tilde{p}_G(y_t|x_t, \theta),
\]

then one form for the correction is

\[
\epsilon_T(x_T, y_T, \theta) = p(y_T|x_T, \theta)/\tilde{p}_G(y_T|x_T, \theta).
\]

(3.4)

While in our case \( \tilde{p}(x_T|y_{1:T}, \theta) \) is not derived as the marginal of a joint approximation \( \tilde{p}(x_{1:T}|y_{1:T}, \theta) \) once INLA is no longer available, it is true that both the EKF and
UKF are derived from assuming approximating Gaussian likelihoods for the observation equation. This suggests that a correction of the form of Equations 3.3 and 3.4 may still be valid for some Gaussian approximation \( \tilde{p}_G(y_T|x_T, \theta) \) to the likelihood.

We propose a \( \tilde{p}_G(y_T|x_T, \theta) \) whose mean and variance are computed by propagating \( \tilde{p}(x_T|y_{1:T}, \theta) \) through the observation function \( f \) of the state-space representation; in this way, the non-linearity between \( x_t \) and \( y_t \) in the observation equation is taken into account. This is done by computing the sigma points from the unscented Kalman filter, denoted \( \tilde{x}_1, \ldots, \tilde{x}_n \) (see the Appendix for their derivation), and if necessary samples of the exogenous and observation error variables \( u_1, \ldots, u_n \) and \( v_1, \ldots, v_n \) from their respective distributions given \( \theta \). Then the mean and variance of \( \tilde{p}(y_T|x_T, \theta) \) are the sample mean and variance of \( f(\tilde{x}_1, u_1, v_1, \theta), \ldots, f(\tilde{x}_n, u_n, v_n, \theta) \).

The additional computational load of this correction is dominated by choosing sigma points. This must be repeated for each \( \theta \) in the discrete grid.

4. DYNAMIC GRID

The last time \( T = T_{\text{INLA}} \) when \( \tilde{p}(\theta|y_{1:T}) \) is computed by INLA will give us a grid \( \Theta_{T_{\text{INLA}}} \) whose scale and orientation is based on an estimate of the curvature of \( \tilde{p}(\theta|y_{1:T}) \) at a mode \( \theta^* = \arg\max_\theta \tilde{p}(\theta|y_{1:T_{\text{INLA}}}) \). Specifically, INLA uses the eigenvectors of the inverse Hessian of \( \log \tilde{p}(\theta|y_{1:T_{\text{INLA}}})|_{\theta = \theta^*} \) for the coordinate axes of \( \Theta_{T_{\text{INLA}}} \). Along axis \( k \) there are \( n_{\text{grid},k}^i \) (\( i = 1,2 \)) points in direction \( i \) from the mode; INLA determines \( n_{\text{grid},k}^i \) by how quickly the approximation decays from the mode. Along each axis the points
are equally spaced. We write $\theta$ in this basis, and so the grid $\Theta_{T_{\text{INLA}}}$ can be written as:

$$
\Theta_{T_{\text{INLA}}} = \{(\theta_1^* \pm \delta_1 n_1, \ldots, \theta_K^* \pm \delta_K n_K) | \\
 n_k \in \{0, 1, \ldots, n_{i_{\text{grid},k}}^i\}, i \in (1, 2)\},
$$

where $\delta_k$ is the grid spacing along the $\theta_k$ axis, determined by the curvature at $\theta^*$. Thus there are $\prod_{k=1}^{K}(n_{i_{\text{grid},k}}^1 + n_{i_{\text{grid},k}}^2 + 1)$ points in $\Theta_{T_{\text{INLA}}}$. Typical values of $n_{i_{\text{grid},k}}^i$ are 6 or 7.

For the rest of the paper, the superscript for $n_{i_{\text{grid},k}}^i$ will be dropped so that complication with respect to notations can be avoided. It will be assumed hereon that for a particular axis, an equal number of points are chosen on either direction of the mode.

### 4.1 Checking the Grid

For $T > T_{\text{INLA}}$, fast alternative methods of updating the grid must be determined. Our proposal is to check the values of $\tilde{p}(\theta|y_{1:T})$ every $T_{\text{update}}$ observations after $T = T_{\text{INLA}}$ to determine if the grid is still appropriate.

We adopt a fairly simple approach to grid updating that nevertheless is fast and appears to be sufficiently accurate in the examples that we consider later. First, the orientation of the axes on which $p(\theta|y_{1:T})$ is computed is not changed from that of the last evaluation with INLA, that is we assume that by $T = T_{\text{INLA}}$ we have established a basis that will continue to be efficient for the discrete representation of the posterior.
density. Then the marginal densities for each component \( \theta_k \) are computed:

\[
\tilde{p}(\theta_{kj}|y_{1:T}) = \left( \frac{\sum_{i_1,\ldots,i_K=-n_{\text{grid},k}}^{n_{\text{grid},k}} \tilde{p}(\theta_1^* + i_1 \delta_1, \ldots, \theta_{kj}^*, \ldots, \theta_K^* + i_K \delta_k|y_{1:T})}{\prod_{i=1}^{K} \delta_i} \right) \times \
\]

Changes to the grid are considered when the support of the marginals is not adequately covered by the grid by one of the following situations:

1. If a marginal at either the upper and lower grid point values, \( \tilde{p}(\theta_k^* + n_{\text{grid}} \delta_k|y_{1:T}) \) or \( \tilde{p}(\theta_k^* - n_{\text{grid}} \delta_k|y_{1:T}) \), is more than a certain proportion of the value at the marginal’s mode \( \max_j \tilde{p}(\theta_{kj}|y_{1:T}) \) then an extra point will be added to the \( k \)th coordinate axis at that end e.g. at \( \theta_k^{\text{add}} = \theta_k^* + (n_{\text{grid}} + 1)\delta_k \) or \( \theta_k^{\text{add}} = \theta_k^* - (n_{\text{grid}} + 1)\delta_k \) respectively.

This adds a \( K - 1 \) dimensional hyperplane

\[
\Theta_k^{\text{add}} = \{(\theta_1, \ldots, \theta_{k-1}, \theta_k^{\text{add}}, \theta_{k+1}, \ldots, \theta_K) | \theta_i = \theta_i^* \pm \delta_i n_i, n_i = 0, 1, \ldots, n_{\text{grid},k}, \text{ for } i \neq k\} \tag{4.1}
\]

to the grid and so

\[
\Theta_{T+1} = \Theta_T \cup \Theta_k^{\text{add}}.
\]

This is referred to as an external point addition.
2. Similarly, if either $\tilde{p}(\theta_k^* + n_{\text{grid}}\delta_k | y_{1:T})$ or $\tilde{p}(\theta_k^* - n_{\text{grid}}\delta_k | y_{1:T})$ is less than a certain very small proportion of $\max_j \tilde{p}(\theta_{kj} | y_{1:T})$ then $\theta_k^{\text{delete}} = \theta_k^* + n_{\text{grid}}\delta_k$ or $\theta_k^{\text{delete}} = \theta_k^* - n_{\text{grid}}\delta_k$ are removed from the $k$th coordinate axis. This removes a $K - 1$ dimensional hyperplane

$$\Theta_k^{\text{delete}} = \{(\theta_1, \ldots, \theta_{k-1}, \theta_k^{\text{delete}}, \theta_{k+1}, \ldots, \theta_K)\}$$

$$\theta_i = \theta_i^* \pm \delta_i n_i, \quad n_i = 0, 1, \ldots, n_{\text{grid}}, \text{ for } i \neq k$$

of points from the grid and so

$$\Theta_{T+1} = \Theta_T - \Theta_k^{\text{delete}}.$$

This is referred to as an external point deletion.

3. If the slope of the marginal between two existing points, relative to its mode,

$$\left| (\tilde{p}(\theta_{kj} | y_{1:T}) - \tilde{p}(\theta_{kj-1} | y_{1:T})) / \max_j \tilde{p}(\theta_{kj} | y_{1:T}) \right|,$$

is larger than a certain threshold value then an extra point is added to the $k$th axis at $\theta_k^{\text{add}} = (\theta_{kj} + \theta_{kj-1})/2$. This adds a $K - 1$ dimensional hyperplane

$$\Theta_k^{\text{add}} = \{(\theta_1, \ldots, \theta_{k-1}, \theta_k^{\text{add}}, \theta_{k+1}, \ldots, \theta_K)| \theta_i = \theta_i^* \pm \delta_i n_i,$$

$$n_i = 0, 1, \ldots, n_{\text{grid}}, \text{ for } i \neq k\}$$

(4.2)
and

$$\Theta_{T+1} = \Theta_T \cup \Theta_{k}^{\text{add}}.$$  

This is referred to as an internal point addition.

Figure 4 shows examples of external and internal addition. It may be that a point is added to more than one axis at one time, which is permitted as long as the computation time is acceptable. At one time it is possible that both external and internal additions/deletions are made.

4.2 Computing $\tilde{p}(\theta|\textbf{y}_{1:T})$ at new points in $\Theta_k^{\text{add}}$

Linear interpolation and extrapolation of $\log(\tilde{p}(\theta|\textbf{y}_{1:T}))$, along the axis where a point is added, is used as it is fast and our experience so far is that it has reasonable performance.

For an external addition, consider $\theta^{\text{add}} \in \Theta_k^{\text{add}}$ as defined in Eq. 4.1 Then $\tilde{p}(\theta^{\text{add}}|\textbf{y}_{1:T+1})$ is approximated by linearly interpolating $\tilde{p}(\theta^{\text{add}}|\textbf{y}_{1:T})$ along the $k$th co-ordinate axis. Let $\theta^{\text{add},1}$ and $\theta^{\text{add},2}$ be $\theta^{\text{add}}$ but with the $k$th component replaced by $\theta^* + n_{\text{grid}} \delta_k$ and $\theta^* + (n_{\text{grid}} - 1) \delta_k$ respectively e.g. $\theta^{\text{add},1}$ and $\theta^{\text{add},2}$ are the two nearest points on the existing grid along the $k$th axis to $\theta^{\text{add}}$. Then

$$\log(\tilde{p}(\theta^{\text{add}}|\textbf{y}_{1:T+1})) = 2 \log(\tilde{p}(\theta^{\text{add},1}|\textbf{y}_{1:T})) - \log(\tilde{p}(\theta^{\text{add},2}|\textbf{y}_{1:T})). \quad (4.3)$$

For an internal addition, consider $\theta^{\text{add}} \in \Theta_k^{\text{add}}$ as defined in Eq. 4.2 Then $\tilde{p}(\theta^{\text{add}}|\textbf{y}_{1:T+1})$ is also approximated by linearly interpolating along the $k$th co-ordinate axis. Let $\theta^{\text{add},1}$
Figure 1: Examples in 2 dimensions where points are added or deleted from the grid with • denoting the existing grid points and + denoting the ones that are added or deleted. From top to down: adding internal points along 2 axes, adding external points along 2 axes, deleting external points along 2 axes.
and $\theta^{\text{add},2}$ be $\theta^{\text{add}}$ but with the $k$th component replaced by $\theta_{k,j-1}$ and $\theta_{k,j}$ respectively e.g. $\theta^{\text{add},1}$ and $\theta^{\text{add},2}$ are the points in the existing grid on either side of $\theta^{\text{add}}$ along the $k$th axis. Then

$$\log(\tilde{p}(\theta^{\text{add}}|y_{1:T+1})) = \left(\log(\tilde{p}(\theta^{\text{add},1}|y_{1:T})) + \log(\tilde{p}(\theta^{\text{add},2}|y_{1:T}))\right)/2. \quad (4.4)$$

---

Specify $T_{\text{INLA}}$ and $T_{\text{update}}$;

let $T = 1$;

repeat:

observe $y_T$;

if $T \leq T_{\text{INLA}}$:

compute discrete grid $\Theta_T$ using INLA;

compute $\tilde{p}(\theta|y_{1:T})$ for $\theta \in \Theta_T$ using INLA;

else:

if $(T - T_{\text{INLA}})$ mod $T_{\text{update}} = 0$:

update grid $\Theta_T$ using method of Section 4.;

else:

$\Theta_T = \Theta_{T-1}$;

end if;

for each $\theta \in \Theta_T$:

compute $\tilde{p}(x_T|y_{1:T-1}, \theta)$ and $\tilde{p}(x_T|y_{1:T}, \theta)$;

update $\tilde{p}(\theta|y_{1:T})$ from $\tilde{p}(\theta|y_{1:T-1})$ by Equation 2.3;

end if;

$T = T + 1$;

end repeat;

---

Table 1: The sequential parameter estimation algorithm.
5. EXAMPLES AND COMPARISON WITH ALTERNATIVE APPROACHES

Our method is implemented on three examples and compared to INLA (an offline method) and an ordinary particle filter. Average performance is measured across many replications of simulated data. To keep the computation time comparison fair, all methods were implemented in R. Our own code was written to implement INLA (e.g. we did not use the R-INLA package). For the particle filter, the R package from [King et al. 2010] was used. The model performances were compared using three different measures, as described below. In all these cases $T_{\text{INLA}}$ lay between 20 and 100.

5.1 Performance measures

The performance of our proposed method is compared to other approaches using three metrics:

Mahalanobis distance: This is used as a measure to judge the accuracy of the estimates of the parameters in the model [Mahalanobis 1936]. For some vector $\theta$ (the true parameter value used to simulate the data), the Mahalanobis distance is defined as:

$$D(\hat{\theta}) = \sqrt{(\hat{\theta} - \theta)^T S_\theta^{-1} (\hat{\theta} - \theta)},$$

(5.1)
where $\hat{\theta}$ and $S_\theta$ are a posterior point estimate (such as mean or mode) and posterior covariance (calculated from the approximate posterior density on the grid).

**Marginal coverage proportion:** Although a frequentist inference property, coverage is a useful measure of whether the posterior uncertainty reflects accurately the error in estimating $\theta$. Marginal coverage is the proportion of times that the true value of a component of the parameter lies within a region of given marginal posterior probability. Over $M$ runs of the inference procedure on distinct data sets, the marginal coverage proportion for a component $\theta_i$ is:

$$\left( \frac{\sum_{m=1}^{M} \mathbb{I}(\theta_{L,m} \leq \theta_i \leq \theta_{H,m})}{M} \right)$$

where $\mathbb{I}$ is the *indicator function* and $\theta_{L,m}$ and $\theta_{H,m}$ are, for example, 95% posterior probability bounds on the marginal distribution of $\theta_i$ for run $m$.

**Computation time:** The time to compute the posterior approximation is also recorded.

### 5.2 Linear Spatial Gaussian model

The statistical model in this example has been assumed to be of the form:

$$y_t = x_{t-1} \mathbf{1} + \eta_t \quad (5.2)$$

$$x_t = \phi x_{t-1} + \epsilon_t \quad (5.3)$$
where $\epsilon_t \sim N(0, \sigma^2_{\text{err}})$ and $\eta_t \sim \mathcal{MNN}(0, \Sigma)$. We assume $y_t$ to be of dimension 3. The covariance matrix $\Sigma$ is also assumed to be dependent on a single unknown parameter:

$$\Sigma = \sigma_{\text{obs}} \Sigma^*.$$ 

The entries of $\Sigma^*$ are of the following type:

$$\Sigma_{ij} = \begin{cases} 
1 & \text{if } i = j, \\
\exp(-rd(i,j)) & \text{if } i \neq j,
\end{cases}$$

where $r > 0$ and $d(i,j)$ is some measure of distance between nodes $i$ and $j$. $\Sigma$ defines the well-known Gaussian spatial process (Matérn, 1986).

Data has been generated by fixing the values at $\phi = 0.35, \sigma^2_{\text{obs}} = 0.004$ and $\sigma^2_{\text{sys}} = 0.035$, and the form of $\Sigma^*$ is known and fixed. A re-parameterization is done to the parameters $\rho_{\text{Obs}}$ and $\rho_{\text{Sys}}$. The algorithm is run with $T_{\text{INLA}} = 20$.

The Kalman filter has been used for optimal filtering at each step, since the system and observation equations are linear. The above simulation has been replicated 50 times. The approximate mode of each of the parameters along with the approximate 95% probability bounds are shown in Figure 2.

Figure 2 shows that our method emulates the ordinary particle filter in terms of accuracy of estimates but Figure 5 shows that it is faster to compute.
Figure 2: Plots (a), (b) and (c) represent trace plots showing trajectories of the averaged approximate mode and approximate 95% probability bounds of the posteriors of $\phi$, $\rho_{Sys}$ and $\rho_{Obs}$ respectively. The light grey line displays the true parameter value. Plot (d) compares the performance of our method with ordinary SMC. Plot (e) does the same using coverage proportion.
5.3 Nonlinear Gaussian model

The following non-linear Gaussian model is used:

\[ y_t = \theta x_t^2 + v_t, \quad (5.4) \]
\[ x_{t+1} = 4 + \sin(\omega \pi t) + \phi x_t + w_t. \quad (5.5) \]

Here \( v_t \sim \mathcal{N}(0, \sigma_{\text{obs}}^2), \ w_t \sim \mathcal{N}(0, \sigma_{\text{sys}}^2) \) and \( \omega \) is assumed to be known.

Data is generated by setting the values at \( \phi = 0.7, \theta = 2, \sigma_{\text{obs}}^2 = 0.35 \) and \( \sigma_{\text{sys}}^2 = 0.0001 \) respectively. A re-parameterization, as in the previous example, is also implemented for this problem. The algorithm is run with \( T_{\text{INLA}} = 50. \)

Figure 3 depicts the trajectories of the mode and 95\% probability bounds of the approximate posterior marginals, as well as the true values of the parameters. The algorithm has been successful in estimating the parameters, except for \( \rho_{\text{sys}} \) which is clearly outside the bounds. Figure 3. One can see that, on an average, our method out-performs the SMC at all time points in terms of accuracy. Figure 5 shows that our method has comparable computation time. INLA is the most accurate but considerably slower.
Figure 3: Trace plots of mode and 95% probability bounds for the parameters $\phi, \theta, \rho_{\text{Sys}}$ and $\rho_{\text{Obs}}$ are shown in Plots (a), (b), (c) and (d) respectively. Our method is compared to SMC in terms of accuracy in Plots (d) and (e).
5.4 Non-Gaussian model

The last example features the application of our proposed algorithm on a non-Gaussian observation equation: T

\[ y_t \sim \text{Poisson (exp}(6 + x_t)\text{)}, \quad (5.6) \]
\[ x_{t+1} = \phi x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2). \quad (5.7) \]

The model above, has been put in an approximating state space form by the following:

\[ y_t = \exp(6 + x_t)\eta_t, \quad \eta_t \sim \log-\mathcal{N}(0, \exp(6 + x_t)), \quad (5.8) \]
\[ x_{t+1} = \phi x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2). \quad (5.9) \]

Data is generated from Equations (5.6) and (5.7) by fixing the values of the parameters \( \phi = 0.35 \) and \( \sigma^2 = 0.2 \). As in the previous examples a re-parameterization is done to the parameters:

\[ \kappa = \logit \left( \frac{\phi + 1}{2} \right) = 0.73 \text{ and } \log \sigma^2 = \log(\sigma_x^2) = -1.6. \]

The algorithm is run with \( T_{\text{INLA}} = 100. \)

The trajectory plot of the estimates and probability bounds in Figure 4 demonstrates the success of our algorithm in estimating the parameters. It is interesting to note the adaptability of the dynamic grid process in this example where one can see that INLA
Figure 4: Plots (a) and (b) show trajectories of the mode and 95% marginal probability bounds for parameters $\kappa$ and $\log(\sigma_x^2)$. Performance comparison between our method and SMC is done in Plots (c) and (d).

A comparison with SMC, as shown in Figure 4, shows that the particle filter is slightly more accurate in this example, although our method is slightly faster (see Figure 5). Again, INLA is the most accurate and the slowest.
6. CONCLUDING REMARKS

A method of fast sequential parameter estimation for dynamic state space models has been proposed and compared to two alternatives: the integrated nested Laplace approximation, and a particle filter. In all the examples that we consider, INLA proved the most accurate but the slowest. Our method achieved similar accuracy to the particle filter but was always faster. It is also worth noting that our method does not suffer from issues of degeneracy that affect the particle filter (Doucet and Johansen [2009]).

Our method has several appealing properties. It can be applied to models with a non-Gaussian latent process, and so is more general than INLA. However it does make use of INLA for initialization and therefore does perform better when the process is Gaussian. It is a very flexible framework in that any filtering and prediction algorithm, not just those used here, can be plugged in as the approximations \( \tilde{p}(x_T|y_{1:T-1}, \theta) \) and \( \tilde{p}(x_T|y_{1:T}, \theta) \). The dynamic grid updating by log-linear interpolation is fast, simple and seems to work well, at least in the examples described in the paper; many alternative interpolation methods could be implemented once this approach fails to be sufficiently accurate. Marginal filtering and prediction are quick to compute from the approximations as

\[
p(x_T|y_{1:T}) \approx \sum_j p(x_t|y_{1:T}, \theta_j)\tilde{p}(\theta_j|y_{1:T}),
\]

\[
p(y_{T+1}|y_{1:T}) \approx \sum_j p(y_{T+1}|y_{1:T}, \theta_j)\tilde{p}(\theta_j|y_{1:T}).
\]
Finally, the algorithm is trivially parallelizable over the grid; the computation of \( \tilde{p}(\theta_j | y_{1:T}) \) at each \( \theta_j \) is completely independent.

The principal disadvantage of the approach is that it is restricted to models with a relatively small number of fixed parameters. Where the number of parameters of interest are small enough, and others are considered to be nuisance parameters, one approach is that the latter can be estimated once after some initial observations and then considered to be fixed. Another potential solution is application of Rao-Blackwellization (Doucet et al., 2000).

As well as ways to increasing the number of parameters that the method can be applied to, another topic for future work is to improve accuracy. We observe that our method is entirely forward-looking, and never revises past approximations in light of new data. The backward filtering equations provide an obvious means to do this; see Baum et al. (1970) or Scott (2002) for their application in a Bayesian setting. This provides opportunities for greater accuracy at some increased computational cost.

Finally there is a need to develop asymptotic properties related to the convergence of the filter. While some idea of the accuracy of posterior of \( \theta | Y_{1:t} \) seems to be directly related to the dimension of the latent process as shown by Rue et al. (2009) for INLA, it needs to be extended in a sequential setting. We feel that the consistency properties of our filter are completely dependent on that of the state filtering mechanism. But this needs to be looked into as future work.
Appendix A: UNSCENTED KALMAN FILTER

UPDATING EQUATIONS

The goal of this method is to compute Gaussian approximations \( \tilde{p}(x_t|y_{1:t-1}, \theta) \) and \( \tilde{p}(x_t|y_{1:t}, \theta) \). This is done by sequentially updating the means and variances of the approximations. The method is best described via the state-space representation of the model:

\[
\begin{align*}
y_t &= f(x_t, u_t, v_t, \theta); \\
x_t &= g(x_{t-1}, w_t, \theta).
\end{align*}
\]

The method starts with an initial mean \( \bar{\chi}_0 \) and variance \( \Sigma_0 \) for \( x_0 \), which can be taken from its prior for example, and approximation parameters \( 0 < \alpha < 1, \beta > 0 \) and \( \kappa > 0 \), whose role will be described later. Letting the dimension of \( x_t \) be \( p \), 2 sets of \( 2p + 1 \) weights \( \omega_{im} \) and \( \omega_{iv} \) are computed as:

\[
\begin{align*}
\omega_{m0} &= \lambda/(p + \lambda); \\
\omega_{i0} &= \omega_{m0} + (1 - \alpha^2) + \beta; \\
\omega_{mi} = \omega_{vi} &= 1/2(p + \lambda), \quad i = -p, \ldots, p, p \neq 0,
\end{align*}
\]

where \( \lambda = \kappa - (1 - \alpha^2)p \).
Updating $\tilde{p}(x_t|y_{1:t-1}, \theta)$: Assuming that at time $t - 1$ there is a mean $\bar{x}_{t-1}$ and variance $\Sigma_{t-1}$ for $\tilde{p}(x_{t-1}|y_{1:t-1}, \theta)$, the method proceeds to update these at time $t$ by defining a set of $2p + 1$ points $\{x_{ti}|i = -p, \ldots, p\}$ centred around $\bar{x}_{t-1}$:

$$
\begin{align*}
\chi_{t0} &= \bar{x}_{t-1}; \\
\chi_{ti} &= \bar{x}_{t-1} + \text{sign}(i)\sqrt{p + \lambda(\sqrt{\Sigma_{t-1}})}i, \\
&i = -p, \ldots, -1, 1, \ldots, p,
\end{align*}
$$

where $(\sqrt{\Sigma_{t-1}})_{i,\cdot}$ refers to the $i$th row of a matrix square root of $\Sigma_{t-1}$ e.g. Cholesky, LU decomposition. The Gaussian approximation to the prediction, $\tilde{p}(x_t|y_{1:t-1}, \theta)$, then has mean

$$
\bar{g}_t = \sum_{i=-p}^{p} \omega_m g(x_{ti})
$$

and variance

$$
\Sigma_x = W_t + \sum_{i=-p}^{p} \omega_v (g(x_{ti}) - \bar{g}_t)(g(x_{ti}) - \bar{g}_t)^T,
$$

where $W_t$ is the variance of the system error $w_t$.

Updating $\tilde{p}(y_t|x_{1:t-1}, \theta)$: A mean and variance for the prediction of $y_t$ given $x_{1:t-1}$ and $\theta$ are then computed as:

$$
\bar{f}_t = \sum_{i=-p}^{p} \omega_m f(x_{ti})
$$
and

\[ \Sigma_{yt} = V_t + \sum_{i=-p}^{p} \omega_{vi}(f(\chi_{ti}) - \bar{f}_t)(f(\chi_{ti}) - \bar{f}_t)^T, \]

where \(V_t\) is the variance of the observation error \(v_t\).

**Updating \(\tilde{p}(x_t|y_{1:t}, \theta)\):** Finally, on observation of \(y_t\), the Gaussian approximation to the smoothing density \(\tilde{p}(x_t|y_{1:t}, \theta)\) has mean and variance:

\[ \bar{\chi}_t = \bar{\chi}_{t-1} + K_t(y_t - \tilde{f}_t) \]

and

\[ \Sigma_t = \Sigma_{x_t} + K_t \Sigma_{yt} K_t^T, \]

where \(K_t = \Sigma_{x_t,y_t} \Sigma_{yt}^{-1}\) and

\[ \Sigma_{x_t,y_t} = V_t + \sum_{i=-p}^{p} \omega_{vi}(g(\chi_{ti}) - \bar{g}_t)(f(\chi_{ti}) - \bar{f}_t)^T \]

is the covariance between \(x_t\) and \(y_t\) given \(x_{1:t-1}\).

It is seen that the method parameters \(\alpha, \beta\) and \(\kappa\) control various aspects of the approximation. A positive semi-definite covariance matrix is guaranteed by \(\kappa > 0\). The dispersion of the sigma points is controlled by \(\alpha \in (0, 1)\) but should ideally be a small number to avoid sampling non local effects when the nonlinearity is strong. The role of \(\beta > 0\) is to minimize higher order errors if prior knowledge of the distribution of \(x_t\) is available.
Figure 5: These three plots compare our method with ordinary SMC and INLA, both in terms of accuracy of estimation and computation speed for each of the model in our examples. INLA has been implemented since it is expected to be most accurate method among the three.

Appendix B: OTHER PLOTS
Figure 6: Linear model: Trace plots showing trajectories of the approximate mode and 95% probability bounds of the posteriors of $\phi(0.001)$, $\rho_{sys}(1)$ and $\rho_{obs}(10)$ respectively.

Figure 7: Trajectories of the mode and 95% marginal probability bounds are plotted for $\kappa$ and $\log(\sigma_x^2)$ in Plots (a) and (b).
References

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

Andrieu, C., A. Doucet, and V. B. Tadic (2005). On-line parameter estimation in general state-space models. In IEEE (Ed.), *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 332–337.

Baum, L., T. Petire, G. Souiles, and N. Weiss (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics* 41, 164–171.

Cseke, B. and T. Heskes (2010). Improving posterior marginal approximations in latent gaussian models. In Y. W. Teh and M. Titterington (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Volume 9, pp. 121 – 128. JMLR Workshop and Conference Proceedings.

Doucet, A., N. de Freitas, K. Murphy, and S. Russell (2000). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, San Francisco, CA, pp. 176–183. Morgan Kaufmann.
Doucet, A., N. D. Freitas, and N. Gordon (2001). *Sequential Monte Carlo methods in practice*. Springer.

Doucet, A. and A. M. Johansen (2009). A tutorial on particle filtering and smoothing: fifteen years later. In *Oxford Handbook of Nonlinear Filtering*. Oxford University Press.

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

Hamilton, J. D. (1986). State-space models. In R. F. Engle and D. McFadden (Eds.), *Handbook of Econometrics*, Volume 4, Chapter 50, pp. 3039–3080. Elsevier.

Haykin, S. (2001). *Kalman filtering and neural networks*. Wiley-Interscience.

Haykin, S., K. Huber, and Z. Chen (2004). Bayesian sequential state estimation for mimo wireless communications. *Proceedings of the IEEE* 92(3), 439–454.

Ionides, E. L., A. Bhadra, Y. Atchadé, and A. King (2011). Iterated filtering. *Annals of Statistics* 39(3), 1776–1802.

Julier, S. J. and J. Uhlmann (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, Florida*, pp. 182–193.
Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering* 1(82), 35–45.

Kantas, N., S. S. Singh, and J. Maciejowski (2009). An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *Proceedings IFAC System Identification (SySid) Meeting*.

King, A. A., E. L. Ionides, C. M. Bretó, S. Ellner, B. Kendall, H. Wearing, M. J. Ferrari, M. Lavine, and D. C. Reuman (2010). *pomp: Statistical inference for partially observed Markov processes (R package)*.

Mahalanobis, P. C. (1936). On the generalised distance in Statistics. *Proceedings of the National Institute of Sciences of India* 2(1), 49–55.

Matérn, B. (1986). *Spatial Variation*. Springer-Verlag.

McElhoe, B. A. (1966). An assessment of the navigation and course corrections for a manned flyby of Mars or Venus. *IEEE Transactions on Aerospace and Electronic Systems* 2, 613–623.

Meinhold, R. J. and N. D. Singpurwalla (1983). Understanding the Kalman filter. *The American Statistician* 37(2), 123–127.

Poyiadjis, G., A. Doucet, and S. S. Singh (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika* 98(1), 65–80.
Rabiner, L. R. (1989, February). A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, Volume 77, pp. 257–286.

Ristic, B., S. Arulampalam, and N. Gordon (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House.

Rue, H., S. Martino, and N. Chopin (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* 71, 319–392.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: recursive computing in the 21st Century. *Journal of the American Statistical Association* 97(457), 337–351.

West, M. and J. Harrison (1997). *Bayesian forecasting and dynamic models* (Second ed.). Springer series in Statistics. Springer.