Algorithms & Computationally Intensive Inference Reading Group
"Introduction to Particle Filtering" Discussion - Notes

Murray Pollock
Address: C1.03, Dept. Of Statistics, University of Warwick, Coventry, CV4 7AL, UK
email: murray.pollock@warwick.ac.uk

22 October, 2010

1 Hidden Markov models

1.1 Hidden Markov models and state space models in time series analysis

Hidden Markov models (HMMs) and *state space models* (SSMs) are classes of models which have broad applicability to a number of diverse areas such as genome analysis [27], robotics [19], neural networks [11] and financial modelling [29]. The development of these model classes has converged to such an extent that in recent literature the terms are used synonymously, however the foundations of these models can be traced separately back to the 1960s with papers such as those of Kalman [23] and Baum et al. (for instance, [4] and [3]). Advancement of this area has been rapid with, for example, consideration to problems of the control of linear systems in the 1980s (for instance [10] and [17]), sequential Monte Carlo in the 1990s (for instance [12]) and many other research fields. Indeed, in some areas such as time series analysis it forms core reading (see [6]). To avoid confusion throughout the remainder of this report we will consider HMMs in the form introduced in the remainder of this section.

HMMs assume that there is some *latent (hidden or unobserved)* underlying $\mathcal{X}^{\mathbb{N}}$-valued generating process $\{X_t\}_{t\geq 0}$[1] of interest known to satisfy the *Markov property*[2], the realizations of which are used to generate some *observable* $\mathcal{Y}^{\mathbb{N}}$-valued process $\{Y_t\}_{t\geq 0}$ with which we wish to use to infer properties of the latent process.

The Markov property of the latent process is particularly useful as it allows the joint distribution of a sequence of states to be represented as the product of states conditional on a finite number of prior states, for instance, in the case where the Markov property is *first order*[3] (and denoting $x_{a:b} := \{x_a, x_{a+1}, \ldots, x_{b-1}, x_b\}$ (where $0 \leq a \leq b$) and $\mathbb{P}$ as a measure on $(\Omega, \mathcal{F})$ such that $\mathcal{F} \to [0,1]$) then,

$$\begin{aligned}\mathbb{P}(x_{0:t}) &= \mathbb{P}(x_t|x_{0:t-1}) \cdot \mathbb{P}(x_{t-1}|x_{0:t-2}) \cdot \ldots \cdot \mathbb{P}(x_0) \\ &= \mathbb{P}(x_0) \cdot \prod_{i=1}^{t} \mathbb{P}(x_i|x_{i-1}) \end{aligned} \tag{1}$$

Throughout this report we will assume we are considering HMMs with $\mathbb{R}^p$-valued latent processes and $\mathbb{R}^q$-valued observational processes and will further assume the following *state space dynamics* (SSDs) hold for some static parameter $\theta \in \Theta$[4],

- $X_0 \sim \mu_\theta(\cdot)$ - A known *initial probability density*.
- $X_t|(X_{t-1} = x_{t-1}, \ldots, X_0 = x_0) = X_t|(X_{t-1} = x_{t-1}) \sim f_\theta(\cdot|x_{t-1})$ - The *latent process* $\{X_t\}_{t\geq 0}$ is a *Markov chain* with a known probability density *conditional* on the prior latent state.

[1] As a convention throughout this report upper case letterings denote random variables whereas lower case letterings denote their realization.
[2] *Markov property:* The conditional distribution of any future state is dependent only upon a finite set of past states.
[3] *nth order Markov property:* The process is dependent on on the current state and the $n-1$ prior states.
[4] Or, alternatively consider some prior on the parameter.

- $Y_t|(X_t = x_{t-1}, Y_{t-1} = y_{t-1}, \ldots, Y_0 = y_0) = Y_t|(X_t = x_{t-1}) \sim g_\theta(\cdot|x_{t-1})\ \forall k \in \mathbb{N}$ - The *observational process* $\{Y_t\}_{t\geq 1}$ has a probability density which is conditionally independent given the latent process.

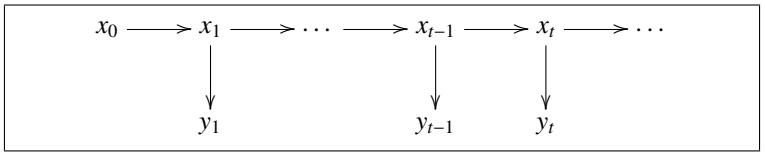


Figure 1: Directed acyclic graph representing a HMM.

This framework is very appealing (for instance) in complex time series problems as the structure of the model allows for a problem to be decomposed into a number of more manageable components where in turn their interactions can be studied (as an example consider a financial application where we may wish to build a model considering various economic indicators such as trend, seasonality and so on).

A canonical illustration of the use of HMMs within time series analysis is to consider the monitoring of some missile launch where information regarding the process of interest (the missile location) is observable only through some other process (a radar output perhaps). Indeed, variations of this and other object tracking problems have been extensively studied (for instance, [34] and [5]).

Typical problems arising in time series include *filtering* (the problem of incorporating observed information to form some probabilistic interpretation of the latent process), *prediction* (the problem of extending known information to make probabilistic interpretations of the future) and *smoothing* (the problem of making improved probabilistic assessments of the latent process at some prior time given all currently observed information), all of which can be tackled within a HMM framework.

The flexible and probabilistic structure of HMMs and the ability to decompose the joint state distribution into a recursive incorporation of additional information provides exact solutions to the time series problems (as shown in §1.2). Furthermore, as state estimates don't have to be recalculated at each time step but instead are being modified with the addition of more information they are particularly suited to tackling *online*[5] problems.

1.2 Recursion problems

Considering the joint density of $X_{0:t}$ and $Y_{1:t}$ then it can be shown by applying conditional probability arguments and the Markov property that this can be expressed simply as the recursion of the known SSDs (where $p_\theta(\cdot)$ denotes some density parameterized by θ),

$$\begin{aligned} p_\theta(x_{0:t}, y_{1:t}) &= p_\theta(y_{1:t}|x_{0:t}) \cdot p_\theta(x_{0:t}) \\ &= p_\theta(y_t|x_{0:t}) \cdot p_\theta(y_{1:t-1}|y_t, x_{0:t}) \cdot p_\theta(x_t|x_{0:t-1}) \cdot p_\theta(x_{0:t-1}) \\ &= g_\theta(y_t|x_t) \cdot p_\theta(y_{1:t-1}|x_{0:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot p_\theta(x_{0:t-1}) \\ &= p_\theta(x_{0:t-1}, y_{1:t-1}) \cdot g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}) = \ldots \\ &= \mu_\theta(x_0) \cdot \prod_{k=1}^{t} \left(g_\theta(y_k|x_k) \cdot f_\theta(x_k|x_{k-1}) \right) \end{aligned} \tag{2}$$

This simple idea of recursion can be easily extended to also consider problems of filtering, prediction and smoothing (as shown in the following sections).

1.2.1 The filtering problem

As outlined in §1.1 when considering *filtering problems* we are interested in incorporating all currently known information from the observational process to form a probabilistic interpretation of the *joint* distribution of the

[5] An *online* problem is one for which sequential information must be processed on arrival without loss of computational efficiency.

latent process. In particular if we again apply conditional probability and Markov property arguments then,

$$\begin{aligned} p_\theta(x_{0:t}|y_{1:t}) &= \frac{p_\theta(x_{0:t}|y_{1:t-1}) \cdot\ p_\theta(y_t|x_{0:t}, y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})} \\ &= \frac{p_\theta(x_{0:t-1}|y_{1:t-1}) \cdot\ p_\theta(x_t|x_{0:t-1}, y_{1:t-1}) \cdot\ g_\theta(y_t|x_t)\, p_\theta(x_{0:t-1}, y_{1:t-1}|x_t, y_t)}{p_\theta(x_{0:t-1}, y_{1:t-1}|x_t) \cdot\ p_\theta(y_t|y_{1:t-1})} \\ &= p_\theta(x_{0:t-1}|y_{1:t-1}) \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot\ g_\theta(y_t|x_t)}{p_\theta(y_t|y_{1:t-1})} \end{aligned} \tag{3}$$

Similarly it may also be of interest to consider *marginal filtering problems* where we consider the distribution of the current latent state given all currently observed information,

$$p_\theta(x_t|y_{1:t}) = \int p_\theta(x_{t-1:t}|y_{1:t})\, \mathrm{d}x_{t-1} \tag{4}$$

$$\begin{aligned} &= \int p_\theta(x_{t-1}|y_{1:t}) \cdot\ p_\theta(x_t|x_{t-1}, y_{1:t})\, \mathrm{d}x_{t-1} \\ &= \int \left(\frac{p_\theta(x_{t-1}|y_{1:t-1}) \cdot\ p_\theta(y_t|x_{t-1})}{p_\theta(y_t|y_{1:t-1})} \right) \cdot \left(\frac{f_\theta(x_t|x_{t-1}) \cdot\ g_\theta(y_t|x_t)}{p_\theta(y_t|x_{t-1})} \right) \mathrm{d}x_{t-1} \\ &= \int p_\theta(x_{t-1}|y_{1:t-1}) \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot\ g_\theta(y_t|x_t)}{p_\theta(y_t|y_{1:t-1})}\, \mathrm{d}x_{t-1} \end{aligned} \tag{5}$$

Noting that (4) is simply marginalization then the remainder of the argument follows using conditional probability and the Markov property as before.

As such we have shown that within a HMM framework that both the *joint* and *marginal filtering problems* are simply the recursion of a *normalized*[6] product of knowns - the solution to the *prior* filtering problem, a *predictive step* $f_\theta(x_t|x_{t-1})$ capturing the hidden process evolution and an *update step* incorporating the latest information $g_\theta(y_t|x_t)$.

Note that in general (unless otherwise stated) we will consider the marginal filtering problem in this report (however the results are extendable).

1.2.2 The prediction problem

The *prediction problem* (as outlined in §1.1) is that of projecting currently observed information which jointly forms the current solution to the *filtering problem* to make a probabilistic interpretation of some future latent state. In particular, we are interested in the marginal distribution of the latent state in p time-steps (X_{t+p}) and as such find (applying the arguments used before),

$$\begin{aligned} p_\theta(x_{t+p}|y_{1:t}) &= \int p_\theta(x_{t:t+p}|y_{1:t})\, \mathrm{d}x_{t:t+p-1} \\ &= \int p_\theta(x_t|y_{1:t}) \cdot\ p_\theta(x_{t+1:t+p}|x_t, y_{1:t})\, \mathrm{d}x_{t:t+p-1} \\ &= \int p_\theta(x_t|y_{1:t}) \cdot\ p_\theta(x_{t+1:t+p}|x_t)\, \mathrm{d}x_{t:t+p-1} = \dots \\ &= \int p_\theta(x_t|y_{1:t}) \cdot \prod_{k=1}^{p} f_\theta(x_{t+k}|x_{t+k-1})\, \mathrm{d}x_{t:t+p-1} \end{aligned} \tag{6}$$

This representation of the prediction problem is intuitively appealing as it can be thought of simply as the current filtered distribution propagated p time-steps into the future using the known latent SSDs.

[6] It is common within sequential Monte Carlo to use arguments of *normalization* which is simply a note that if a probability density can be found to some constant of proportionality then in principle the exact density can be found by using the fact that a probability density integrates to 1 (which in many cases is a non-trivial problem).

1.2.3 The smoothing problem

Finally, the *smoothing problem* is that of using all information from the observed process currently available to make revised estimates of the marginal filtering problem for states of the latent process at some prior time (with the intuition that we would expect to achieve better estimates of the latent state than in the filtering problem as more (future) information is available). In particular, we are interested in the marginal distribution of the latent process X_k (for some $k \in (0, t)$) conditional on $Y_{1:t}$,

$$\begin{aligned}
p_\theta(x_k|y_{1:t}) &= \int p_\theta(x_{k:k+1}|y_{1:t})\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot\ p_\theta(x_k|x_{k+1}, y_{1:t})\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot \frac{p_\theta(x_k|x_{k+1}, y_{1:k}) \cdot\ p_\theta(y_{k+1:t}|x_{k:k+1}, y_{1:k})}{p_\theta(y_{k+1:t}|x_{k+1}, y_{1:k})}\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot\ p_\theta(x_k|x_{k+1}, y_{1:k})\,\mathrm{d}x_{k+1} \\
&= \int p_\theta(x_{k+1}|y_{1:t}) \cdot \frac{p_\theta(x_k|y_{1:k}) \cdot\ p_\theta(x_{k+1}|x_k, y_{1:k})}{p_\theta(x_{k+1}|y_{1:k})}\,\mathrm{d}x_{k+1} \\
&= p_\theta(x_k|y_{1:k}) \cdot \int \frac{p_\theta(x_{k+1}|y_{1:t}) \cdot\ f_\theta(x_{k+1}|x_k)}{\int p_\theta(x_k|y_{1:k}) \cdot\ f_\theta(x_{k+1}|x_k)\,\mathrm{d}x_k}\,\mathrm{d}x_{k+1}
\end{aligned} \tag{7}$$

Again the solution is intuitively appealing as it is simply the initial estimate of the filtering problem with a backwards recursive modification representing currently observed information (the terms of which are known).

The extension of the smoothing problem to a *multivariate setting* is often of interest in that instead of limiting the problem to the univariate marginal $p_\theta(x_k|y_{1:t})$ where $k \in [0, t)$ we may instead be interested in problems considering periods of time - for instance consider $p_\theta(x_{k_1}, x_{k_2}, \ldots, x_{k_q}|y_{1:t})$ where $q < t$ and $k_1, \ldots, k_q \in [0, t]$ - however this is computationally far more challenging.

1.3 Limitations of state space models with reference to the the Kalman filter

Unfortunately, even though in §1.2 explicit representations for various recursion problems were found, implementation is still not always possible. Calculation of the normalizing constant $p_\theta(y_t|y_{1:t-1})$ in the filtering problem inhibits direct implementation of the results in all but a few specific cases. Furthermore, there is considerable difficulty in dealing with high dimensional or highly non-linear problems due to lack of analytic tractability.

A number of particular instances where direct implementation of the methodology is possible exist (namely those illustrated by Vidoni [38]), however by far the most studied is the formulation of the SSDs given within the context of the *Kalman filter* which was developed primarily by Kalman in the 1960s ([23] and [22]). In this instance the SSDs are Gaussian which allows the filtering problem to be solved without the need to find the normalizing constant.

Although a HMM with Gaussian SSDs may seem limiting various extensions have been made which generalize the methodology to more complicated HMMs (those which are non-linear and non-Gaussian) by modifying them to fit them within the framework of the Kalman filter. Examples of (among many) such extensions include the *extended Kalman filter* (eKF), which in cases where the SSDs are non-linear but sufficiently smooth considers a linearized version of them (see [2]) and the *unscented Kalman filter* (UKF) which for highly non-linear SSDs involves deterministically sampling from the prior density (drawing so called *sigma points*) propagating the samples through the model SSDs and reconstructing the mean and covariance of the posterior (see [21]).

A brief illustration of the Kalman filter is considered as it neatly illustrates the use of HMMs in time series applications and is useful when considering both the *optimal importance function* (see §2.6.1) and the *auxiliary particle filter* (see §2.7) which are introduced later (for reference [31] provides a fuller account of the Kalman filter).

Suppose we are interested in the *marginal filtering problem* (4) where the following SSDs hold $\forall t \geq 1$ (denoting matrices by upper case emboldened letters),

$$X_0 \sim MVN(m_0, \mathbf{C}_0) \tag{8}$$

$$X_t|(X_{t-1} = x_{t-1}) \sim MVN(\mathbf{A}x_{t-1}, \mathbf{V}_t) \tag{9}$$

$$Y_t|(X_t = x_t) \sim MVN(\mathbf{B}x_t, \mathbf{W}_t) \tag{10}$$

Following an induction argument we wish to show that,

$$X_t|(Y_{1:t} = y_{1:t}) \sim MVN(m_t, \mathbf{C}_t) \tag{11}$$

Trivially for $t = 0$ we have $X_0 \sim MVN(m_0, \mathbf{C}_0)$ which agrees with (8). Furthermore assume we know the solution to the marginal filtering problem at time $t-1$ $\Big(X_{t-1}|(Y_{1:t-1} = y_{1:t-1}) \sim MVN(m_{t-1}, \mathbf{C}_{t-1})\Big)$.

Now, note that if we re-express the marginal filtering problem introduced in (4) we have,

$$p_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t) \cdot\ p_\theta(x_t|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})} \propto\ g_\theta(y_t|x_t) \cdot\ p_\theta(x_t|y_{1:t-1}) \tag{12}$$

Considering a marginalization of $p_\theta(x_{t-1}|y_{1:t-1})$ and introducing the known SSDs then note that we have a convolution of two Gaussian distributions which can be shown to also be a Gaussian distribution[7]. As the distribution of interest is Gaussian it can be simply characterized by finding its first two moments (using the *tower rule*),

$$\mathbb{E}(X_t|Y_{1:t-1}) = \mathbb{E}(\mathbb{E}(X_t|X_{t-1}, Y_{1:t-1})|Y_{1:t-1}) = \mathbf{A}m_{t-1} \tag{16}$$

$$\begin{aligned}\mathbb{V}ar(X_t|Y_{1:t-1}) &= \mathbb{E}(\mathbb{V}ar(X_t|X_{t-1}, Y_{1:t-1})|Y_{1:t-1}) + \mathbb{V}ar(\mathbb{E}(X_t|X_{t-1}, Y_{1:t-1})|Y_{1:t-1}) \\ &= \mathbf{V}_t + \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T = \mathbf{P}_t\end{aligned} \tag{17}$$

$$X_t|(Y_{1:t-1} = y_{1:t-1}) \sim MVN(\mathbf{A}m_{t-1}, \mathbf{P}_t) \tag{18}$$

We now have representations for both terms in the re-expressed marginal filtering problem (12) and as such we find,

$$\begin{aligned}p_\theta(x_t|y_{1:t}) &\propto\ g_\theta(y_t|x_t) \cdot\ p_\theta(x_t|y_{1:t-1}) \\ &\propto \exp\bigg(-\frac{1}{2}\Big(y_t - \mathbf{B}x_t\Big)^T\mathbf{W}_t^{-1}\Big(y_t - \mathbf{B}x_t\Big)\bigg) \cdot \exp\bigg(-\frac{1}{2}\Big(x_t - \mathbf{A}m_{t-1}\Big)^T\mathbf{P}_t^{-1}\Big(x_t - \mathbf{A}m_{t-1}\Big)\bigg) \\ &\propto \exp\bigg(-\frac{1}{2}\Big(x_t^T\mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B}x_t - 2x_t^T\mathbf{B}^T\mathbf{W}_t y_t\Big)\bigg) \cdot \exp\bigg(-\frac{1}{2}\Big(x_t^T\mathbf{P}_t^{-1}x_t - 2x_t^T\mathbf{P}_t^{-1}\mathbf{A}m_{t-1}\Big)\bigg) \\ &\propto \exp\bigg(-\frac{1}{2}\Big(x_t^T(\mathbf{P}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B})x_t - 2x_t^T(\mathbf{P}_t^{-1}\mathbf{A}m_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t)\Big)\bigg)\end{aligned} \tag{19}$$

[7] In order to show the convolution is Gaussian it is sufficient to show this to some normalizing in constant. In particular,

$$p_\theta(x_t|y_{1:t-1}) = \int p_\theta(x_{t:t-1}|y_{1:t-1})\,\mathrm{d}x_{t-1} \tag{13}$$

$$= \int f_\theta(x_t|x_{t-1}) \cdot\ p_\theta(x_{t-1}|y_{1:t-1})\,\mathrm{d}x_{t-1} \tag{14}$$

$$\begin{aligned}&\propto \int \exp\bigg(-\frac{1}{2}\Big(x_t - \mathbf{A}x_{t-1}\Big)^T\mathbf{V}_t^{-1}\Big(x_t - \mathbf{A}x_{t-1}\Big)\bigg)\exp\bigg(-\frac{1}{2}\Big(x_{t-1} - m_{t-1}\Big)^T\mathbf{C}_{t-1}^{-1}\Big(x_{t-1} - m_{t-1}\Big)\bigg)\mathrm{d}x_{t-1} \\ &\propto \int \exp\bigg(-\frac{1}{2}\Big(x_{t-1}^T\mathbf{A}^T\mathbf{V}_t^{-1}\mathbf{A}x_{t-1} - 2x_{t-1}^T\mathbf{A}^T\mathbf{V}_t^{-1}x_t\Big)\bigg)\exp\bigg(-\frac{1}{2}\Big(x_{t-1}^T\mathbf{C}_{t-1}^{-1}x_{t-1} - 2x_{t-1}^T\mathbf{C}_{t-1}^{-1}m_{t-1}\Big)\bigg)\mathrm{d}x_{t-1} \\ &= \int \exp\bigg(-\frac{1}{2}\Big(x_{t-1}^T(\mathbf{A}^T\mathbf{V}_t^{-1}\mathbf{A} + \mathbf{C}_{t-1}^{-1})x_{t-1} - 2x_{t-1}^T(\mathbf{A}^T\mathbf{V}_t^{-1}x_t + \mathbf{C}_{t-1}^{-1}m_{t-1})\Big)\bigg)\mathrm{d}x_{t-1}\end{aligned} \tag{15}$$

Note that (15) is simply the integrated unnormalized density of the MVN distribution.

Noting that (19) has the form of a Gaussian density then we find,

$$\begin{aligned}
\mathbf{C}_t^{-1} &= \mathbf{P}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B} \\
&= (\mathbf{V}_t + \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T)^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B} \qquad (20) \\
m_t &= \mathbf{C}_t(\mathbf{P}_t^{-1}\mathbf{A}m_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t) \\
&= \mathbf{C}_t((\mathbf{V}_t + \mathbf{A}\mathbf{C}_{t-1}\mathbf{A}^T)^{-1}\mathbf{A}m_{t-1} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t) \qquad (21)
\end{aligned}$$

As a consequence we have found the desired result,

$$X_t|(Y_{1:t} = y_{1:t}) \sim MVN(m_t, \mathbf{C}_t) \qquad (22)$$

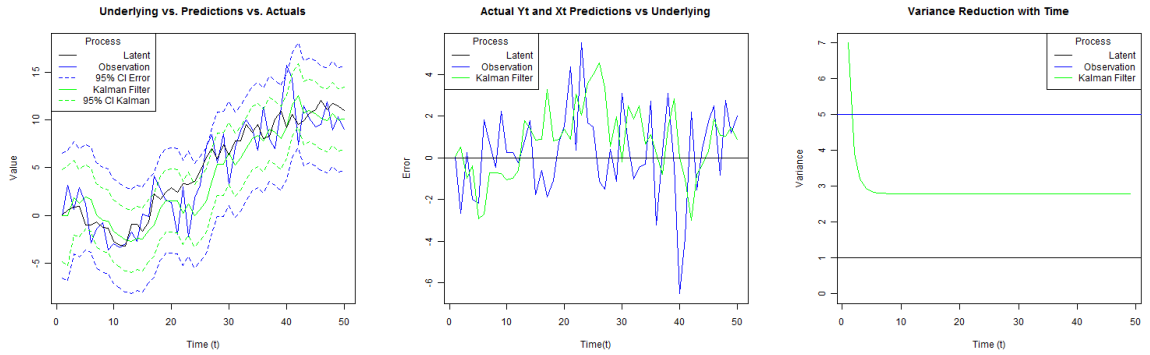
Intuitively the Kalman filter can be thought of rather simply as the recursion of a prior Gaussian state and Gaussian information to give a Gaussian posterior to be used as the next prior.

1.3.1 Kalman filter numerical illustration

In order to illustrate the Kalman filter the results of §1.3 the Kalman filter is applied to the first 50 observations of data simulated from the following SSDs,

$$X_0 \sim N(0, 6) \qquad (23)$$
$$X_t|(X_{t-1} = x_{t-1}) \sim N(x_{t-1}, 1) \qquad (24)$$
$$Y_t|(X_t = x_t) \sim N(x_t, 5) \qquad (25)$$

A realization of one such simulation is shown in *Figure 2*. In particular, *Figures 2(a) and 2(b)* show that although the Kalman filter manages a reduction in the observational noise and achieves a better estimate of the latent process than the observation process, it is still strongly influenced by extreme observations.



(a) Kalman filter estimate of latent process over time.

(b) Latent process adjusted Kalman filter estimate over time.

(c) Kalman filter state estimate variance over time.

Figure 2: Kalman filter applied to HMM with Gaussian SSDs.

Considering *Figure 2(c)* note that the latent process and observational state estimate variances are constant as per the model SSDs (23 - 25) and in this case the Kalman filter manages to reduce the noise from the observational process providing tighter state estimate variance. Furthermore, it is worth noting that in the case where there is zero noise from the observational process then the lowest possible variance in the Kalman filter estimate that could be achieved is that of the latent process (and as such in this example it is expected the observational process adds some margin onto this).

The expected improvement in the Kalman filter state estimate variance over time due to the sequential integration of more information is also clearly shown (easily found using (17) and recursively applying (20)), indeed it can be shown that the Kalman filter is the minimum variance unbiased estimator as it attains the Cramer-Rao lower bound.

2 Sequential Monte Carlo methods

2.1 Motivation

From §1 we have established that despite the appealing recursive nature of HMMs which address a large class of problems, particularly problems for which sequential information must be processed on arrival without loss of computational efficiency (*online problems*), difficulties with estimating the normalizing constant in the filtering problem (3) and extension to problems of a high dimensional and/or highly non-linear nature limit the extent of their application to specific exceptions (such as [23] and [38]).

The most natural approach to circumvent problems of state estimation would be to apply MCMC techniques, however these are of limited use in online problems as the computational cost in estimating states isn't fixed over time (and the state has to be re-estimated at all times). Indeed, the acquisition of additional information over time *increases* the computational cost. A number of (somewhat unsatisfactory) solutions exist attempting to circumvent this problem such as using MCMC techniques for some subset of the data of a fixed size capping the computational cost but reducing the integrity of the data, or alternatively making some of the approximations to the Kalman filter discussed in §1.3 (which depending on the degree of true non-linearity and non-Gaussianity in the HMM perform arbitrarily poorly).

Particle filtering methods (the application of sequential Monte Carlo techniques to HMMs) were introduced in 1993 by *Gordon et al.* [15] in the form of the *bootstrap filter* and subsequently a great deal of theory has been developed in order to improve, justify and optimize the basic idea they presented. In essence particle filters aim to take advantage of the inherent structure and time correlation of HMMs by providing posterior state estimates using recyclable draws (*particles*) from the prior state distribution which are propagated and re-weighted in time using importance sampling techniques and appropriate referencing to the known SSDs and observations.

The development of particle filtering methodology is rather disjointed as it was driven primarily by applications and the rigorous justification of the methodology was developed afterwards. Within this report particle filters are considered within the context of importance sampling as in this manner it is clear to see that they are variations on a single idea. It is worth noting that a number of more detailed tutorials and texts exist (such as [13], [14], [30] and [12]).

2.2 Importance sampling

To introduce *importance sampling* (IS) lets say we are interested in some density f which is intractable. Now, suppose there is some other density g which is tractable and has the property that $f(x) > 0 \Rightarrow g(x) > 0 \; \forall x$ then we can circumvent the intractability as follows (where S denotes the entire state space),

$$\begin{aligned} I = \mathbb{E}_f(h(X)) &= \int_S h(x) \cdot f(x) \, \mathrm{d}x \\ &= \int_S \frac{h(x) \cdot f(x)}{g(x)} \cdot g(x) \, \mathrm{d}x \\ &= \mathbb{E}_g\Big(\frac{h(X) \cdot f(X)}{g(X)}\Big) \end{aligned} \tag{26}$$

Note that we have reformulated the intractable f in terms of the tractable g (known as the *proposal density* or alternatively the *importance function*). The requirement for the condition that $f(x) > 0 \Rightarrow g(x) > 0 \; \forall x$ can lead to issues with convergence and as such the stricter condition that there exists some constant $M > 0$ s.t. $f(x) < M \cdot g(x)$ $\forall x$ is used in practice (in essence ensuring the proposal density has sufficiently heavy tails) and ensures that all

parts of the proposal density are generated sufficiently regularly in order to have convergence to the target density. Now, an IS estimate of I can be obtained by drawing from g,

$$\begin{aligned}\hat{I} &= \frac{1}{N}\sum_{i=1}^{N}\frac{h(x^{(i)})\cdot f(x^{(i)})}{g(x^{(i)})} \\ &= \frac{1}{N}\sum_{i=1}^{N} h(x^{(i)})\cdot w(x^{(i)})\end{aligned} \tag{27}$$

where,

$$w(x^{(i)}) = \frac{f(x^{(i)})}{g(x^{(i)})} \tag{28}$$

Intuitively this can be thought of as simulating f by instead simulating g and weighting the samples according to their relative density at the sampled point ($w(\cdot)$ - the *importance weighting function*). In essence the sample $\{x^{(i)}, w^{(i)}\}_{i=1}^{N}$ can jointly be thought of as a discrete approximation of f.

Relating this idea to the recursion problems discussed in §1.2 where we only know the target density up to some constant of proportionality (C) we find we are trying to evaluate problems of the form,

$$I = \mathbb{E}_f(h(X)) = \int_S C\cdot h(x)\cdot \tilde{f}(x)\,\mathrm{d}x \tag{29}$$

Now letting $\bar{w}(x^{(i)}) = C\cdot w^*(x^{(i)}) = \dfrac{C\cdot\tilde{f}(x^{(i)})}{g(x^{(i)})}$ then applying (27) we have,

$$\begin{aligned}\mathbb{E}_f(C) &= \int_S C\cdot f(x)\,\mathrm{d}x \\ &= \int_S \bar{w}(x)\cdot g(x)\,\mathrm{d}x \\ &= \mathbb{E}_g(\bar{w}(X))\end{aligned} \tag{30}$$

and as such,

$$\begin{aligned}\mathbb{E}_f(h(X)) &= \mathbb{E}_g(h(X)\cdot\bar{w}(X)) \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\frac{h(x^{(i)})\cdot w^*(x^{(i)})}{C} \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\frac{h(x^{(i)})\cdot w^*(x^{(i)})}{\frac{1}{N}\sum_{j=1}^{N} w^*(x^{(j)})} \qquad (31) \\ &= \sum_{i=1}^{N} h(x^{(i)})\cdot w(x^{(i)}) \qquad (32)\end{aligned}$$

noting that (31) is possible using *Slutsky's lemma*[8] where,

$$w(x^{(i)}) = \frac{w^*(x^{(i)})}{\sum_{j=1}^{N} w^*(x^{(j)})} \tag{33}$$

Specifically IS removes the need to calculate the constant of proportionality as provided we can calculate the *unnormalized importance weights* $\left(\{w^*(x^{(i)})\}_{i=1}^{N}\right)$ then they can be adjusted to produce *normalized weights* $\left(\{w(x^{(i)})\}_{i=1}^{N}\right)$, circumventing the primary problem in applying the HMM recursions (§1.2) at a cost of introducing *slight bias* (due to (31)) which diminishes as the number of samples increases.

[8] *Slutsky's lemma:* Let X_n converge in distribution to some random element X and let $\{Y_t\}$ convergent to a constant c then the following hold - $X_n + Y_n \xrightarrow{\mathcal{D}} X + c$; $Y_n X_n \xrightarrow{\mathcal{D}} cX$; $\dfrac{X_n}{Y_n} \xrightarrow{\mathcal{D}} \dfrac{X}{c}$

2.3 Bayesian sequential importance sampling

Relating §2.2 to the filtering problem (3) we would like to sequentially update the prior estimated state ($p_\theta(x_{0:t-1}|y_{1:t-1})$) incorporating new observations (y_t) to estimate the current state ($p_\theta(x_{0:t}|y_{1:t})$) without the need to estimate the normalizing constant ($p_\theta(y_t|y_{1:t-1})$).

Now suppose there exists some normalized importance function $\pi_\theta(x_{0:t}|y_{1:t})$ from which we can independently draw samples $\{x_{0:t}^{(i)}, 1/N\}_{i=1}^N$ which has support including that of the state posterior. Furthermore, lets choose the importance function such that it allows recursive updates in time as subsequent observations become available,

$$\pi_\theta(x_{0:t}|y_{1:t}) = \pi_\theta(x_t|x_{0:t-1}, y_{1:t}) \cdot \pi_\theta(x_{0:t-1}|y_{1:t-1}) \tag{34}$$

Now consider the filtering problem (3) then in a similar manner to the results of §2.2 we find,

$$\begin{aligned}
\mathbb{E}_{p_\theta(x_{0:t}|y_{1:t})}(h(x_{0:t})) &= \int_S h(x_{0:t}) \cdot p_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \\
&= \int_S h(x_{0:t}) \cdot \frac{p_\theta(x_{0:t}|y_{1:t})}{\pi_\theta(x_{0:t}|y_{1:t})} \cdot \pi_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \\
&\propto \int_S h(x_{0:t}) \cdot \frac{p_\theta(x_{0:t-1}|y_{1:t-1})}{\pi_\theta(x_{0:t-1}|y_{1:t-1})} \cdot \frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{\pi_\theta(x_t|x_{0:t-1}, y_{1:t})} \cdot \pi_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \\
&= \mathbb{E}_{\pi_\theta(x_{0:t}|y_{1:t})}(h(x_{0:t}) \cdot w_t) \\
&\approx \frac{1}{N}\sum_{i=1}^N h(x_{0:t}^{(i)}) \cdot w_t^{(i)}
\end{aligned} \tag{35}$$

where the importance weights are defined as follows,

$$w_t^{(i)} = \frac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}} \tag{36}$$

$$w_t^{*(i)} = w_{t-1}^{*(i)} \cdot \frac{f_\theta(x_t|x_{t-1}^{(i)}) \cdot g_\theta(y_t|x_t)}{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})} \tag{37}$$

Note, that in order to use this result samples are made from the importance function $\pi_\theta(x_{0:t}|y_{1:t})$, however, as the importance function has been constructed in a recursive manner in (34) then (without loss of generality) each of the sample state paths which form the discrete approximation of the prior state ($\{x_{0:t-1}^{(i)}\}_{i=1}^N$) can be updated using the *conditional importance function* ($\pi_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t})$) to produce samples for the posterior state ($x_{0:t}^{(i)} := (x_{0:t-1}^{(i)}, x_t^{(i)})$). Updating (37) and normalization (36) of the importance weights ($w_t^{(i)}$) jointly with the updated samples ($x_{0:t}^{(i)}$) form a discrete approximation of the posterior state (where δ is the Dirac delta function),

$$p_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} \approx \hat{p}_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} = \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}) \tag{38}$$

The *sequential importance sampling* (SIS) algorithm (*Algorithm 1*) is the natural extension of the above methodology.

It is clear that the SIS algorithm is straightforward to implement provided the SSDs are available, however we are left with the problem of choosing a suitable conditional importance function such that the sample paths which constitute the prior state estimate are propagated into areas of the sample space which are sufficiently likely that they can be used to construct the posterior estimate (further discussed in §2.6). Selection of this conditional importance function is a key combat against the inherent problem in particle filters of *degeneracy* in the importance weights (introduced in §2.4).

Algorithm 1 Sequential importance sampling (SIS) algorithm

Initialization Step ($t = 0$):

for all $i = 1, \dots, N$ **do**
  sample $x_0^{(i)} \sim \mu_\theta(x_0)$
  set $w_0^{(i)} = 1/N$
end for
set $\hat{p}_\theta(x_0)(\mathrm{d}x_0) = \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$

Observation Update Step ($t \geq 1$):

for all $t = 1, 2, \dots$ **do**
  for all $i = 1, \dots, N$ **do**
    sample $x_t^{(i)} \sim \pi_\theta(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})$
    set $x_{0:t}^{(i)} := (x_{0:t-1}^{(i)}, x_t^{(i)})$
    evaluate weights $w_t^{*(i)} = w_{t-1}^{(i)} \cdot \dfrac{f_\theta(x_t^{(i)} | x_{t-1}^{(i)}) \cdot\ g_\theta(y_t | x_t^{(i)})}{\pi_\theta(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})}$
    normalize weights $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$
  end for
  set $\hat{p}_\theta(x_{0:t} | y_{1:t})\, \mathrm{d}x_{0:t} = \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$
end for

2.4 Degeneracy

The variance of the sample path importance weights increases as the observational process further evolves which has the implication that state estimates become progressively more reliant on a small number sample paths of particularly high weighting (or potentially a single sample path), producing poor resolution of the state estimate while simultaneously expending considerable computational power updating paths which have negligible contribution to the state estimate. This problem is known as *degeneracy* of the importance weights, and can be shown to be true for all choices of recursive importance function (for a more formal account see [26]),

$$\begin{aligned}
\mathbb{V}ar(w_t^*) &= \mathbb{V}ar\left(w_{t-1}^* \cdot \frac{f_\theta(x_t | x_{t-1}) \cdot\ g_\theta(y_t | x_t)}{\pi_\theta(x_t | x_{0:t-1}, y_{1:t})} \right) \\
&= \mathbb{V}ar\left(\mathbb{E}\left(w_{t-1}^* \cdot \frac{f_\theta(x_t | x_{t-1}) \cdot\ g_\theta(y_t | x_t)}{\pi_\theta(x_t | x_{0:t-1}, y_{1:t})} | x_{0:t-1}, y_{1:t-1} \right) \right) + \mathbb{E}\left(\mathbb{V}ar\left(w_{t-1}^* \cdot \frac{f_\theta(x_t | x_{t-1}) \cdot\ g_\theta(y_t | x_t)}{\pi_\theta(x_t | x_{0:t-1}, y_{1:t})} | x_{0:t-1}, y_{1:t-1} \right) \right) \\
&\geq \mathbb{V}ar\left(w_{t-1}^* \cdot \mathbb{E}\left(\frac{f_\theta(x_t | x_{t-1}) \cdot\ g_\theta(y_t | x_t)}{\pi_\theta(x_t | x_{0:t-1}, y_{1:t})} \mid x_{0:t-1}, y_{1:t-1} \right) \right) \\
&= \mathbb{V}ar(w_{t-1}^*)
\end{aligned} \tag{39}$$

Resampling is the intuitive response to the problem of degeneracy. In the case where the importance weights are deemed to have degenerated *sufficiently* then new sample paths are drawn from the discrete approximation of the current state which itself is composed of the current sample paths and the importance weights are reset. In essence this is a trade-off - resampling the current state estimate produces in the short term poorer state estimation, however, preserving a set of sample paths with non-negligible weights ensures better future state estimates. *Figure 3* illustrates this process of resampling where at time $t = 3$ the sample paths have degenerated to such an extent that resampling is required and as such the poor sample paths are discarded in favour of stronger sample paths for the future state estimates.

Other issues which arise from using a resampling procedure include being unable to parallelize the process and additionally, as the sample paths are no longer independent, convergence results are more complicated (although a central limit theorem is available [9]).

We have established that the importance weights degenerate over time and as such at some point resampling

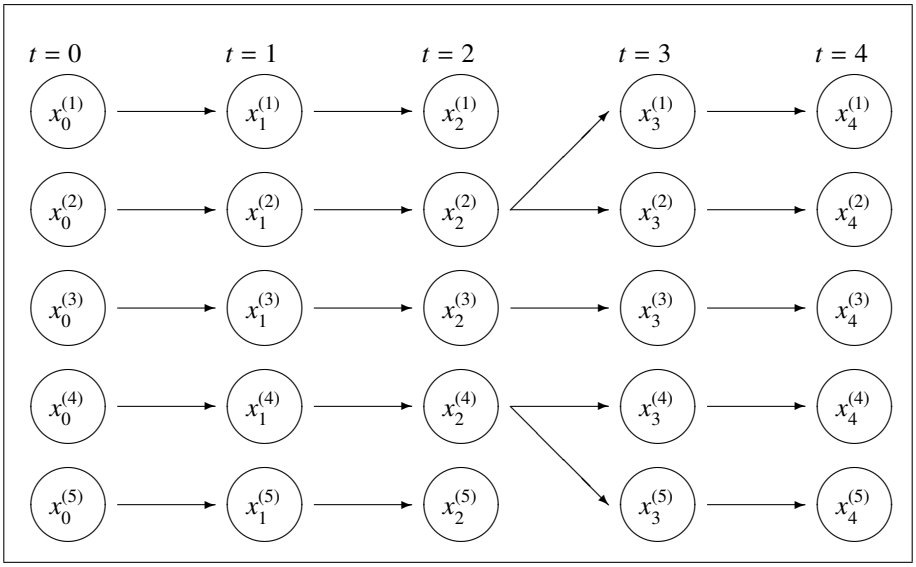


Figure 3: Illustration of sample path SIR algorithm with no resampling other than at $t = 3$.

has to occur. Even though by definition the target density is not available it is useful to think of comparing the degeneracy in the importance weights of the chosen importance function against the degeneracy of some sample from it in order to guage how degenerate the actual sample has become. This intuition leads to the heuristic known as *effective sample size* based developed by Kong in 1992 [25] and extended by Kong, Liu, Wong in 1994 [26]), which in essence is a metric considering the number of samples from the target density which would be required to achieve the same variance as the N samples from the importance function.

Noting the results in the general IS case from §2.2 and considering the problem of evaluating some function h we define our metric as follows (noting the obvious result that simulating from the target is equivalent to having equal weighting in the samples),

$$N_{EFF} := N \cdot \frac{\mathbb{V}ar_{p_\theta(x_{0:t}|y_{1:t})}(h(X))}{\mathbb{V}ar_{\pi_\theta(x_{0:t}|y_{1:t})}(w(X) \cdot h(X))} \tag{40}$$

Kong, Liu, Wong 1994 [26] apply the delta method using the first two moments of w and h which results in the approximation (41),

$$\begin{aligned} N \cdot \frac{\mathbb{V}ar_{p_\theta(x_{0:t}|y_{1:t})}(h(X))}{\mathbb{V}ar_{\pi_\theta(x_{0:t}|y_{1:t})}(w(X) \cdot h(X))} &\approx \frac{N}{1 + \mathbb{V}ar_{\pi_\theta(x_{0:t}|y_{1:t})}(w(X))} && (41) \\ &= \frac{N}{\left(\mathbb{E}_{\pi_\theta(x_{0:t}|y_{1:t})}(w(X))\right)^2 + \mathbb{V}ar_{\pi_\theta(x_{0:t}|y_{1:t})}(w(X))} \\ &= \frac{N}{\mathbb{E}_{\pi_\theta(x_{0:t}|y_{1:t})}((w(X))^2)} \\ &\approx \frac{1}{\sum_{i=1}^{N} (w_t^{(i)})^2} && (42) \\ &= \widehat{N_{EFF}} && (43) \end{aligned}$$

Whenever the *effective sample size* ($\widehat{N_{EFF}}$) falls below some chosen threshold (N_{TH}) the sample paths are resampled. The choice of the threshold is very much up to the user who should be aware of the trade-off made (and mentioned earlier) between increasing variance in current state estimates for better future state estimates.

2.5 Resampling

As we have now concluded that resampling is a necessary evil we are now left with the problem of ensuring that resampling is conducted as efficiently as possible. *Multinomial resampling* is considered first as it was the original method used in the 1993 Gordon *et al.* paper [15] and is very intuitively appealing, however the weaknesses of this method will be explored along with alternate popular resampling methodologies.

2.5.1 Multinomial resampling

Intuitively, in order to draw a single sample path from the empirical state estimate $\hat{p}_\theta(x_{0:t}|y_{1:t})$ constructed from the sample set $\{x_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ one could simply draw randomly $x_{0:t}^{(i)}$ with probability $w_t^{(i)}$. Extending this idea to the case where we want to draw N samples (i.e. to ensure states are estimated with a stable number of particles at all times) then this is equivalent to considering the number of *offspring* $O_t^{(i)}$ of each sample path (i.e. the number of times each sample path is used in the new approximation), as following a *multinomial distribution* (hence the name *multinomial resampling* [15]), resulting in the following resampled state estimate,

$$\bar{p}_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} = \sum_{i=1}^{N} \frac{O_t^{(i)}}{N} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}) \tag{44}$$

Incorporation of the threshold requirement and multinomial resampling within the SIS Algorithm (Algorithm 1) results in what is known as the *sequential importance resampling* (SIR) algorithm [35] (see *Algorithm 2*).

Algorithm 2 Sequential importance resampling (SIR) algorithm

Initialisation Step ($t = 0$):

for all $i = 1, \ldots, N$ **do**
  sample $x_0^{(i)} \sim \mu_\theta(x_0)$
  set $w_0^{(i)} = 1/N$
  set $\widehat{N_{EFF}} = N$
end for
set $\hat{p}_\theta(x_0)(\mathrm{d}x_0) = \sum_{i=1}^N w_0^{(i)} \cdot \delta_{x_0^{(i)}}(\mathrm{d}x_0)$

Observation Update Step ($t \geq 1$):

for all $t = 1, 2, \ldots$ **do**
  if $\widehat{N_{EFF}} \leq N_{TH}$ **then**
    for all $i = 1, \ldots, N$ **do**
      resample $x_{0:t-1}^{(i)} \sim \hat{p}_\theta(x_{0:t-1}|y_{1:t-1})$
      reset $w_{t-1}^{(i)} = 1/N$
    end for
  end if
  for all $i = 1, \ldots, N$ **do**
    sample $x_t^{(i)} \sim \pi_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t})$
    set $x_{0:t}^{(i)} := (x_{0:t-1}^{(i)}, x_t^{(i)})$
    evaluate weights $w_t^{*(i)} = w_{t-1}^{(i)} \cdot \dfrac{f_\theta(x_t^{(i)}|x_{t-1}^{(i)}) \cdot\ g_\theta(y_t|x_t^{(i)})}{\pi_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t})}$
    normalize weights $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$
    evaluate $\widehat{N_{EFF}} = \dfrac{1}{\sum_{i=1}^N (w_t^{(i)})^2}$
  end for
  set $\hat{p}_\theta(x_{0:t}|y_{1:t})(\mathrm{d}x_{0:t}) = \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$
end for

2.5.2 Evaluating resampling efficiency

Intuitively, resampling increases state estimate variance as the resampled state estimate is merely an approximation of the original state estimate. Furthermore, procedures such as multinomial resampling reduce power in the tails of state estimates as they are inclined to choose paths with high weightings. As such it is worth considering procedures which ensure more of the structure of the original state estimate is retained.

More formally, reconsidering multinomial resampling we find, and wish to retain its unbiasedness property ($\mathbb{E}(O_t^{(i)}|w^{(i)}) = N.w_t^{(i)}\ \forall i \in N$), but we wish to reduce the variance ($\mathbb{V}ar(O_t^{(i)}|w^{(i)})$).

An extensive literature concerning resampling procedures is outlined in [12] including unequal resampled importance weights [24] and partial resampling [8], however the procedures considered within this report are the most popular ones. It is worth noting that there are additionally procedures which seek to minimize *mean square error* as opposed to variance (i.e. are biased) [24].

2.5.3 Systematic resampling

Systematic resampling is a straight-forward procedure introduced by Kitagawa in 1996 [24] that in effect breaks the cumulative density function (CDF) into segments each corresponding to the size of a *fairly weighted* particle ($1/N$). One sample path is then drawn from each of the segments at equally spaced points on the CDF, the first draw in the sequence being uniformly chosen in the first segment to ensure that the procedure doesn't repeatedly miss inherent structure in the CDF.

This modification to the SIR algorithm is achieved by substituting the multinomial resampling step (algorithm observation update step 4-5) by *Algorithm 3*,

Algorithm 3 SIR systematic resampling steps

1: **set** cumulative weights, $C = 0$
2: **set** sample path counter, $j = 1$
3: **sample** $u \sim U(0, 1)$
4: **for all** $i = 1, \ldots, N$ **do**
5: **set** $C = w_{t-1}^{(i)} + C$
6: **while** $\frac{u + j - 1}{N} \leq C$ **do**
7: **set** $\tilde{x}_{0:t-1}^{(j)} = x_{0:t-1}^{(i)}$
8: **set** $j = j + 1$
9: **end while**
10: **end for**

2.5.4 Stratified resampling

Stratified resampling was introduced by Clifford et al. in 1999 [7] and is a procedure that breaks the CDF into equally sized segments inversely proportional to the size of the particle set (N). From each segment one sample is drawn (as opposed to systematically based on the initial drawing). As such the procedure is very similar to stratified resampling (though perhaps less appealing due to the additional randomness in the procedure), and can be carried out as per *Algorithm 4*,

2.5.5 Residual Resampling

Residual resampling is an idea introduced by Higuchi in 1997 [18] and Liu and Chen in 1998 [28] which in essence breaks the resampling procedure into two steps. Firstly, any sample paths which have weighting greater than their *fair share* ($1/N$) are resampled at least as many times as increments thereof. Secondly, the remaining required draws are resampled using multinomial resampling from the sample paths with modified weights taking into account any selection in the first step. The SIR algorithm modification to multinomial resampling step (algorithm observation update step 4-5) is shown in *Algorithm 5*,

Algorithm 4 SIR stratified resampling steps

set cumulative weights, $C = 0$
set samples path counter, $j = 1$
for all $i = 1, \ldots, N$ **do**
  sample $u_i \sim U(0, 1)$
  set $l_i = \dfrac{u_i + i - 1}{N}$
end for
for all $i = 1, \ldots, N$ **do**
  set $C = w_{t-1}^{(i)} + C$
  while $l_j \leq C$ **do**
    set $\tilde{x}_{0:t-1}^{(j)} = x_{0:t-1}^{(i)}$
    set $j = j + 1$
  end while
end for

Algorithm 5 SIR residual resampling steps

for all $i = 1, \ldots, N$ **do**
  if $w_{t-1}^{(i)} \geq 1/N$ **then**
    resample $x_{0:t-1}^{(i)}$ **path** $\lfloor N \cdot w_{t-1}^{(i)} \rfloor$ **times**
    set $w_{t-1}^{(i)} = w_{t-1}^{(i)} - \lfloor N \cdot w_{t-1}^{(i)} \rfloor / N$
  end if
end for
find reserved particle numbers $R = N - \sum_{i=1}^{N} \lfloor N \cdot w_{t-1}^{(i)} \rfloor$
normalize weights $w_{t-1}^{(i)} = \dfrac{w_{t-1}^{(i)}}{\sum_{k=1}^{N} w_{t-1}^{(k)}}$
set $r(x_{0:t-1}|y_{1:t-1})(\mathrm{d}x_{0:t-1}) = \sum_{i=1}^{N} w_{t-1}^{(i)} \cdot \delta_{x_{0:t-1}^{(i)}}(\mathrm{d}x_{0:t-1})$
for all $i = 1, \ldots, (N - R)$ **do**
  resample $x_{0:t-1}^{(i)} \sim r(x_{0:t-1}|y_{1:t-1})$
end for
set $w_{t-1}^{(i)} = 1/N \; \forall i$

Clearly *Algorithm 5* is easily extendable, replacing the multinomial resampling part with some of the alternates discussed would potentially improve it further, however this (as mentioned) is by no means meant to be an exhaustive list of methodologies and as such this has been omitted.

2.6 Importance function selection

As outlined in §2.3 selection of an importance function which propagates the prior sample paths into areas of the state space which are sufficiently likely will ensure the importance weights don't degenerate too quickly. Alternatively this can be thought of as choosing an importance function which minimizes the variance of the importance weights, and applying the result in (37) that $w_t^{*(i)} = w_{t-1}^{*(i)} \cdot \dfrac{f_\theta(x_t|x_{t-1}^{(i)}) \cdot g_\theta(y_t|x_t)}{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})}$ we have,

$$\begin{aligned}
\mathbb{V}ar_{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})}(w_t^{*(i)}) &= \int_S (w_t^{*(i)})^2 \cdot \pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})\,\mathrm{d}x_t - \left(\int_S w_t^{*(i)} \cdot \pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})\,\mathrm{d}x_t\right)^2 \\
&= \int_S (w_{t-1}^{*(i)})^2 \cdot \frac{(g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)}))^2}{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})}\,\mathrm{d}x_t - \left(\int_S w_{t-1}^{*(i)} \cdot g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})\,\mathrm{d}x_t\right)^2 \\
&= (w_{t-1}^{*(i)})^2 \cdot \left(\int_S \frac{(g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)}))^2}{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})}\,\mathrm{d}x_t - \left(\int_S p_\theta(x_t, y_t|x_{t-1}^{(i)})\,\mathrm{d}x_t\right)^2\right) \\
&= (w_{t-1}^{*(i)})^2 \cdot \left(\int_S \frac{(g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)}))^2}{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})}\,\mathrm{d}x_t - (p_\theta(y_t|x_{t-1}^{(i)}))^2\right) \qquad (45)
\end{aligned}$$

2.6.1 Optimal importance function

Selection of the importance function $\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t}) = p_\theta(x_t|x_{t-1}^{(i)}, y_t)$ was first introduced by Zaritskii et al. (1975 [39]) and Akashi and Kumamoto (1977 [1]). It was further shown by Doucet *et al.* in 2000 [13] that it is the *optimal importance function* as it can be shown substituting it into (45) that the variance of the importance weights are reduced to zero (see (46)). In particular,

$$\begin{aligned}
\mathbb{V}ar_{\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t})}(w_t^{*(i)}) &= (w_{t-1}^{*(i)})^2 \cdot \left(\int_S \frac{(g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)}))^2}{p_\theta(x_t|x_{t-1}^{(i)}, y_t)}\,\mathrm{d}x_t - (p_\theta(y_t|x_{t-1}^{(i)}))^2\right) \\
&= (w_{t-1}^{*(i)})^2 \cdot \left(\int_S \frac{(g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)}))^2 \cdot p_\theta(y_t|x_{t-1}^{(i)})}{g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})}\,\mathrm{d}x_t - (p_\theta(y_t|x_{t-1}^{(i)}))^2\right) \\
&= (w_{t-1}^{*(i)})^2 \cdot \left(p_\theta(y_t|x_{t-1}^{(i)}) \cdot \int_S p_\theta(x_t, y_t|x_{t-1}^{(i)})\,\mathrm{d}x_t - (p_\theta(y_t|x_{t-1}^{(i)}))^2\right) \\
&= (w_{t-1}^{*(i)})^2 \cdot \left((p_\theta(y_t|x_{t-1}^{(i)}))^2 - (p_\theta(y_t|x_{t-1}^{(i)}))^2\right) = 0 \qquad (46)
\end{aligned}$$

Furthermore, the importance weights for evaluation in implementation of the SIR Algorithm can be shown to be as follows,

$$\begin{aligned}
w_t^{*(i)} &= w_{t-1}^{*(i)} \cdot \frac{g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})}{p_\theta(x_t|x_{t-1}^{(i)}, y_t)} \\
&= w_{t-1}^{*(i)} \cdot \frac{g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})}{\left(\dfrac{f_\theta(x_t|x_{t-1}^{(i)}) \cdot p_\theta(y_t|x_t, x_{t-1}^{(i)})}{p_\theta(y_t|x_{t-1}^{(i)})}\right)} \\
&= w_{t-1}^{*(i)} \cdot p_\theta(y_t|x_{t-1}^{(i)}) \qquad (47)
\end{aligned}$$

The *optimal importance function* is often not implementable directly as samples are required from the future state

$p_\theta(x_t|x_{t-1}^{(i)}, y_t)$ and additionally, the generally intractable $p_\theta(y_t|x_{t-1}^{(i)})$ has to be evaluated. Techniques to overcome these issues are typically analogous to techniques used for the Kalman filter such as local linearization of the HMM (similar to the extended Kalman filter) and local linearization of the optimal importance function (*using a Taylor expansion*) as illustrated by Doucet et al. [13].

2.6.1.1 Linear Gaussian SSM As discussed in §2.6.1 in order to implement the *optimal importance function* we require access to both $p_\theta(x_t|x_{t-1}^{(i)}, y_t)$ (the importance function) and $p_\theta(y_t|x_{t-1}^{(i)})$ (the importance weight update). In order to illustrate a simple example of this lets consider the linear Gaussian SSM as follows,

$$X_0 \sim MVN(m_0, \mathbf{C}_0) \tag{48}$$
$$X_t|(X_{t-1} = x_{t-1}) \sim MVN(\mathbf{A}x_{t-1}, \mathbf{V}_t) \tag{49}$$
$$Y_t|(X_t = x_t) \sim MVN(\mathbf{B}x_t, \mathbf{W}_t) \tag{50}$$

Considering the importance function notice that $p_\theta(x_t|x_{t-1}^{(i)}, y_t) = p_\theta(x_t|x_{t-1}^{(i)}) \cdot p_\theta(y_t|x_t)$. Now in an analogous fashion to (19) the importance function can be found explicitly as the product of two Gaussian densities (and hence is Gaussian),

$$p_\theta(x_t|x_{t-1}^{(i)}, y_t) \propto \exp\left(-\frac{1}{2}\Big(x_t^T(\mathbf{V}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B})x_t - 2x_t^T(\mathbf{V}_t^{-1}\mathbf{A}x_{t-1}^{(i)} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t)\Big)\right) \tag{51}$$

$$X_t|(X_{t-1} = x_{t-1}^{(i)}, Y_t = y_t) \sim MVN\left((\mathbf{V}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B})^{-1}(\mathbf{V}_t^{-1}\mathbf{A}x_{t-1}^{(i)} + \mathbf{B}^T\mathbf{W}_t^{-1}y_t), (\mathbf{V}_t^{-1} + \mathbf{B}^T\mathbf{W}_t^{-1}\mathbf{B})^{-1}\right) \tag{52}$$

Now considering the importance weight update we find $p_\theta(y_t|x_{t-1}^{(i)})$ $(= \int_S g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})\,\mathrm{d}x_t)$. As a result we have the convolution of two Gaussian distributions which in turn is Gaussian (in a similar manner to (15)). Calculating the first two moments we find,

$$\mathbb{E}(Y_t|X_{t-1} = x_{t-1}^{(i)}) = \mathbb{E}(\mathbb{E}(Y_t|X_t)|X_{t-1} = x_{t-1}^{(i)}) = \mathbb{E}(\mathbf{B}X_t|X_{t-1} = x_{t-1}^{(i)}) = \mathbf{B}\mathbf{A}x_{t-1}^{(i)} \tag{53}$$
$$\begin{aligned}\mathbb{V}ar(Y_t|X_{t-1} = x_{t-1}^{(i)}) &= \mathbb{E}(\mathbb{V}ar(Y_t|X_t)|X_{t-1} = x_{t-1}^{(i)}) + \mathbb{V}ar(\mathbb{E}(Y_t|X_t)|X_{t-1} = x_{t-1}^{(i)}) \\ &= \mathbb{E}(\mathbf{W}_t|X_{t-1}) + \mathbb{V}(\mathbf{B}X_t|X_{t-1} = x_{t-1}^{(i)}) = \mathbf{W}_t + \mathbf{B}^T\mathbf{V}_t\mathbf{B}\end{aligned} \tag{54}$$
$$Y_t|(X_{t-1} = x_{t-1}^{(i)}) \sim MVN(\mathbf{B}\mathbf{A}x_{t-1}^{(i)}, \mathbf{W}_t + \mathbf{B}^T\mathbf{V}_t\mathbf{B}) \tag{55}$$

2.6.1.2 Non-linear Gaussian SSM In a similar fashion to the extension of the Kalman filter to the extended Kalman filter it is worthwhile considering cases where the HMM has non-linear SSDs,

$$X_0 \sim MVN(m_0, \mathbf{C}_0) \tag{56}$$
$$X_t|(X_{t-1} = x_{t-1}) \sim MVN(f(x_{t-1}), \mathbf{V}_t) \tag{57}$$
$$Y_t|(X_t = x_t) \sim MVN(g(x_t), \mathbf{W}_t) \tag{58}$$

Assuming that both f and g are differentiable, then as per [13] and [2] applying a first order approximation of the observation mean we have (letting D represent some constant),

$$\begin{aligned} g(x_t) &\approx g(f(x_{t-1})) + \left.\frac{\partial g(x_t)}{\partial x_t}\right|_{x_t = f(x_{t-1})} (x_t - f(x_{t-1})) \\ &= \left.\frac{\partial g(x_t)}{\partial x_t}\right|_{x_t = f(x_{t-1})} x_t + \left(g(f(x_{t-1})) - \left.\frac{\partial g(x_t)}{\partial x_t}\right|_{x_t = f(x_{t-1})} f(x_{t-1})\right) \\ &=: \mathbf{M}x_t + D \end{aligned} \tag{59}$$

Now considering the approximated optimal importance function $\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t}) := \tilde{p}_\theta(x_t|x_{t-1}^{(i)}, y_t)$ (or perhaps

more intuitively the optimal importance function for the approximated HMM) then in a similar manner to (51) we find,

$$\tilde{p}_\theta(x_t|x_{t-1}^{(i)}, y_t) \propto \exp\bigg(-\frac{1}{2}\Big(x_t^T(\mathbf{V}_t^{-1} + \mathbf{M}^T\mathbf{W}_t^{-1}\mathbf{M})x_t - 2x_t^T(\mathbf{V}_t^{-1}f(x_{t-1}^{(i)}) + \mathbf{M}^T\mathbf{W}_t^{-1}(y_t - D))\Big)\bigg) \tag{60}$$

$$\sim MVN\Big((\mathbf{V}_t^{-1} + \mathbf{M}^T\mathbf{W}_t^{-1}\mathbf{M})^{-1}(\mathbf{V}_t^{-1}f(x_{t-1}^{(i)}) + \mathbf{M}^T\mathbf{W}_t^{-1}(y_t - D)), (\mathbf{V}_t^{-1} + \mathbf{M}^T\mathbf{W}_t^{-1}\mathbf{M})^{-1}\Big) \tag{61}$$

Furthermore, analogous to (53) - (55) for the importance weights we have,

$$\mathbb{E}(Y_t|X_{t-1} = x_{t-1}^{(i)}) = \mathbb{E}(\mathbb{E}(Y_t|X_t)|X_{t-1} = x_{t-1}^{(i)}) = \mathbf{M}f(x_{t-1}^{(i)}) + D \tag{62}$$

$$\mathbb{V}ar(Y_t|X_{t-1} = x_{t-1}^{(i)}) = \mathbb{E}(\mathbb{V}ar(Y_t|X_t)|X_{t-1} = x_{t-1}^{(i)}) + \mathbb{V}ar(\mathbb{E}(Y_t|X_t)|X_{t-1} = x_{t-1}^{(i)}) = \mathbf{W}_t + \mathbf{M}^T\mathbf{V}_t\mathbf{M} \tag{63}$$

$$Y_t|(X_{t-1} = x_{t-1}^{(i)}) \sim MVN(\mathbf{M}f(x_{t-1}^{(i)}) + D, \mathbf{W}_t + \mathbf{M}^T\mathbf{V}_t\mathbf{M}) \tag{64}$$

It should be noted that although many of the techniques employed to the optimal importance function in order to broaden its application are similar to those of the Kalman filter, particle filters have the distinct advantage that the sub-optimality of any given importance function proposed is captured within the state estimates whereas the Kalman filter simply provides an approximation to the Gaussian case without any explicit evaluation of the discrepancy.

2.6.2 Prior importance function

The *prior importance function* (with the selection $\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t}) := f_\theta(x_t|x_{t-1}^{(i)})$) was introduced by Handschin and Mayne in (1969) [16] and is so called as the choice of importance function does not take into account the current observation. It should be noted that the original particle filter proposed in 1993 by Gordon et al. [15] (*the bootstrap filter*), in effect used a prior importance function. The importance weights for SIR algorithm implementation can be shown to be,

$$\begin{aligned} w_t^{*(i)} &= w_{t-1}^{*(i)} \cdot \frac{g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})}{f_\theta(x_t|x_{t-1}^{(i)})} \\ &= w_{t-1}^{*(i)} \cdot g_\theta(y_t|x_t^{(i)}) \end{aligned} \tag{65}$$

The prior importance function is considerably easier to evaluate as the update to the importance weights is a known element of the SSDs, however it suffers from more rapid degeneracy as is it is overly sensitive to the current observation, in effect information regarding the observational process is incorporated one step later than optimal resulting in poor propagation of the prior samples to the posterior state. More formally the results relating to importance weight variance in the optimal case (§2.6.1) show the prior importance function to be sub-optimal. It is on the other hand very intuitively appealing as the new importance weights are simply the prior importance weights adjusted for the new observation.

2.6.3 Fixed importance function

The *fixed importance function* ($\pi_\theta(x_t|x_{0:t-1}^{(i)}, y_{1:t}) := p_\theta(x_t)$) used by Tanizaki (1993 [36]) and Tanizaki and Mariano (1994 [37]) is arguably an even simpler importance function selection, however as it in no way incorporates the observation or indeed the sample path dynamics it performs very poorly. For completeness, the importance weights can be found as follows,

$$w_t^{*(i)} = w_{t-1}^{*(i)} \cdot \frac{g_\theta(y_t|x_t) \cdot f_\theta(x_t|x_{t-1}^{(i)})}{p_\theta(x_t)} \tag{66}$$

2.7 Auxiliary particle filter

Within the SIR algorithm observational outliers (particularly in cases where the observation SSD has large variance) can result in uneven predictive distributions. A brute force solution to this problem is to simply draw more

particles from the prior density, however this can be computationally intensive and potentially infeasible. In addition when resampling, the particles chosen typically constitute large weight to the prior empirical density and as such will be chosen to be updated and included in the posterior empirical density, resulting in poor tail estimation. The low selection probabilities of particles in the tail again dictate that for any reasonable estimation of the tails a large number of particles have to be used.

The solution to this problem is to ensure the proposal importance function targets the observation (as discussed in §2.6), however (noting §2.6.1) the *optimal importance function* is typically unavailable due to reasons of intractability. The *auxiliary particle filter* (APF) proposed by Pitt and Shepard in 1999 [32] (and extended in 2001 [33]) attempts to provide an alternate solution.

The formulation of the auxiliary particle filter shown below attempts to introduce the concept in an intuitive fashion using an *auxiliary variable* as originally proposed by Pitt and Shepard but within the SIR framework (in a similar fashion as [20] and [14]) as doing so has the consequence that a number of the results available for the SIR algorithm can be exploited (for example, convergence results).

Reconsidering the original marginal filtering problem (5) at time t given the marginal filtering problem to time $t-1$ we have,

$$\begin{aligned} p_\theta(x_t|y_{1:t}) &\propto \int p_\theta(x_{t-1}|y_{1:t-1}) \cdot f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t) \,\mathrm{d}x_{t-1} \\ &\approx \sum_{i=1}^{N} w_{t-1}^{(i)} \cdot f_\theta(x_t|x_{t-1}^{(i)}) \cdot g_\theta(y_t|x_t) \end{aligned} \tag{67}$$

Now suppose we were to introduce an *auxiliary variable* K (a variable representing a path selection of one of the N *prior* samples) and instead consider the *auxiliary marginal filtering problem*,

$$p_\theta(x_t, K|y_{1:t}) \propto f_\theta(x_t|x_{t-1}^{(K)}) \cdot g_\theta(y_t|x_t) \tag{68}$$

Note that $g_\theta(y_t|x_t)$ is unavailable, however (68) can instead be simulated using importance sampling with the importance function (where $\phi_t^{(K)} := \mathbb{E}(x_t|x_{t-1}^{(K)})$),

$$\pi_\theta(x_t, K|y_{1:t}) \propto f_\theta(x_t|x_{t-1}^{(K)}) \cdot g_\theta(y_t|\phi_t^{(K)}) \tag{69}$$

and importance weights,

$$\begin{aligned} w(x_t, K) &= \frac{p_\theta(x_t, K|y_{1:t})}{\pi_\theta(x_t, K|y_{1:t})} \\ &= \frac{g_\theta(y_t|x_t)}{g_\theta(y_t|\phi_t^{(K)})} \end{aligned} \tag{70}$$

It is now clear that a sample of (67) can be achieved in a 2-step algorithm by drawing from K such that $\mathbb{P}(K = k) = w_{t-1}^{(k)} \cdot g_\theta(y_t|\phi_t^{(k)})$ (in effect choosing a sample path using its prior weight combined with the expectation of its contribution to the posterior empirical density) and then propagating the sample forward using the SSDs with an adjustment according to the importance weight as per *Algorithm 6* (with the usual resampling steps).

In summary - An intuitive way to consider the APF algorithm is rather than choosing particles according to their prior weights then updating and reweighting them given the observation as per the SIR algorithm, we instead look at the observation and choose prior particles which target the observation. More simply stated, the APF algorithm chooses particles with high posterior rather than prior weighting and as such are more efficient as it removes the redundancy of updating particles which contribute negligible weight to the posterior density estimate.

3 Simulations

In order to illustrate the methods shown in §2 three examples are considered; §3.1 - A HMM with Gaussian SSD (for comparison with the Kalman filter); §3.2 - A non-linear HMM; §3.3 - Bearings-only tracking.

Algorithm 6 Auxiliary particle filter (APF) algorithm

1: **for all** $k = 1, \ldots, N$ **do**
2: **sample** λ_k, where $\mathbb{P}(\lambda_k = i) \propto w_{t-1}^{(i)} \cdot g_\theta(y_t|\phi_t^{(i)})$
3: **set** $x_{0:t}^{(k)} := (x_{0:t-1}^{(\lambda_k)}, x_t^{(k)})$ **where** $x_t^{(k)} \sim \pi_\theta(x_t|x_{t-1}^{(\lambda_k)})$
4: **set** $w_t^{*(k)} := \dfrac{g_\theta(y_t|x_t^{(k)})}{g_\theta(y_t|\phi_t^{(k)})}$
5: **end for**
6: **normalize weights** $w_t^{(i)} = \dfrac{w_t^{*(i)}}{\sum_{j=1}^N w_t^{*(j)}}$
7: **evaluate** $\widehat{N_{EFF}} = \dfrac{1}{\sum_{i=1}^N (w_t^{(i)})^2}$
8: **set** $\hat{p}_\theta(x_{0:t}|y_{1:t})\,\mathrm{d}x_{0:t} = \sum_{i=1}^N w_t^{(i)} \cdot \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t})$

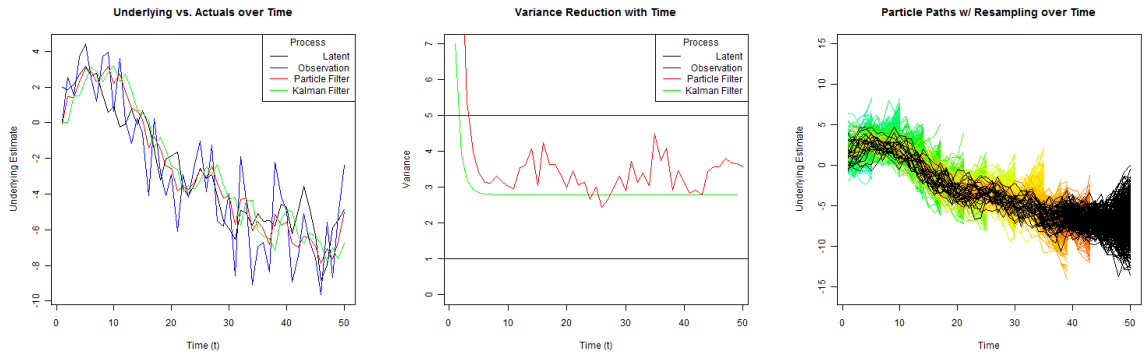
3.1 SSM with Gaussian SSDs

Consider a realization of 50 time steps from the following HMM *initialized at* $m_0 = 0$ with *initial density variance* $V_0 = 6$, *latent process variance* $V_t = 5\ \forall t \geq 1$ and *observational process variance* $W_t = 5\ \forall t$ using a *particle set of size* 500 (the same parameterization as with the Kalman filter illustration of §1.3.1),

$$X_0 \sim N(m_0, \mathbf{C}_0) \tag{71}$$
$$X_t|(X_{t-1} = x_{t-1}) \sim N(ax_{t-1}, \mathbf{V}_t) \tag{72}$$
$$Y_t|(X_t = x_t) \sim N(bx_t, \mathbf{W}_t) \tag{73}$$

3.1.1 Prior importance function

First lets consider applying a particle filter with *prior importance function* (see §2.6.2) and *multinomial* (see §2.5) resampling strategy to the realization from the HMM given in §3.1 along with a Kalman filter applied to the same realization.



(a) Particle filter and Kalman filter estimate of latent process over time.

(b) Particle filter and Kalman filter state estimate variance over time.

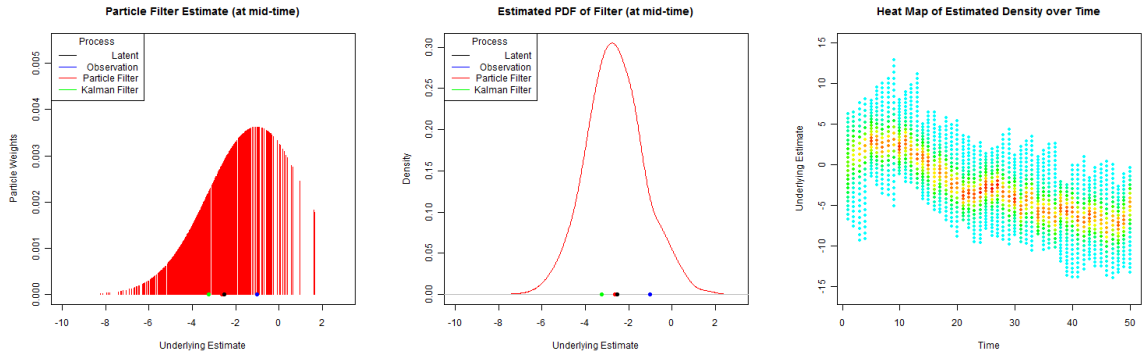
(c) Heat map representing at each time point unique complete sample paths constituting the state estimate.

Figure 4: Prior importance function particle filter applied to HMM with Gaussian SSDs.

Figure 4(a) anecdotally shows the particle filter reduces the noise from the observation process and provides estimates close to that of the Kalman filter. Figure 4(b) confirms the particle filter offers reduction in the variance from the observation process but it doesn't attain the Cramer-Rao lower bound of the Kalman filter. With increased particle set size it would be expected that the particle filter variance would converge to that of the Kalman filter.

Figure 5 illustrates the manner in which the particle filter provides state estimates using a set of particles. In particular: Figure 5(a) shows the weighting given to each of the 500 particles overlayed with points marking the

actual observation, underlying latent process and the Kalman filter estimate at the mid-point of the process; Figure 5(b) shows the state estimate pdf constructed using the particle set and associated importance weightings which encouragingly shows the latent process is close to the mode of the density; Figure 5(c) is a version of Figure 5(b) over all times (where areas of high density are coloured red and this is graduated to blue for areas of low density (scaled by the area of highest density over all time points)), from which we see the filter quickly establishes itself and seems to provide good state estimates.



(a) Individual particles mid-time weightings. (b) Mid-time empirical pdf evaluated using particle set. (c) Empirical pdf evaluated using particle set at all times.

Figure 5: Prior importance function filter particle set representation.

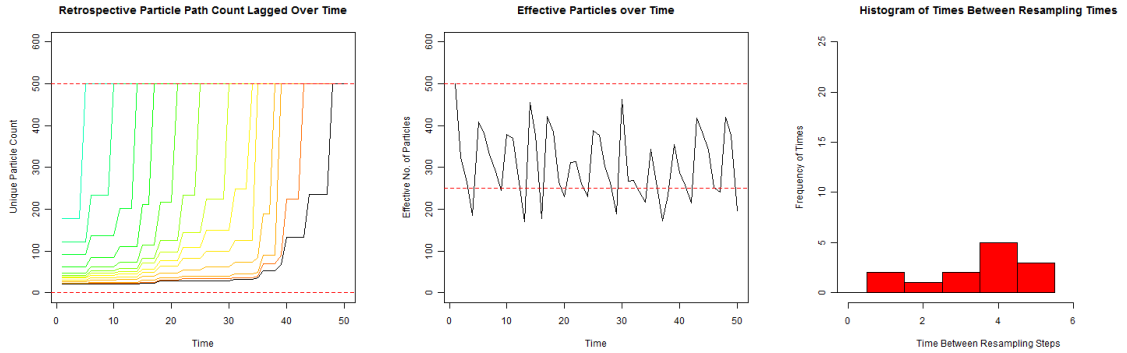
Figure 4(c) shows the particle locations at each time point and their preceding path (black denotes paths which survive to constitute the final state estimate, red denotes paths which *survive* a long time and this is graduated to blue denoting paths which are *eliminated* quickly), in effect illustrating what is happening during resampling. From this we see resampling is frequent which, as discussed, has the effect of eliminating precision in the tails of the state estimate in favour of high density areas. Additionally, it is evident that the number of unique sample paths are rapidly reduced after each resampling indicating strong particle dependence.

Figure 6 is a series of more diagnostic graphs illustrating the anecdotal evidence from Figure 4(c). Figure 6(a) counts backwards from immediately before a resampling (where we have 500 distinct particles) the number of unique sample paths at each time traced to the origin (the colouring with the same representation as before), capturing the rate of degeneracy in time and the diversity of the particle set (which in this case is poor as we have resampled twelve times in 50 time steps and furthermore within a mere 20 time steps the number of unique paths is a tenth of the number of particles). Figure 6(b) shows the *effective sample size* (see §2.4) over time with the accompanying threshold for resampling (set to half of the size of the particle set) which we would like to degenerate slowly and Figure 6(c) shows the lag between resampling points which would ideally be as long as possible. Here we find rapid degeneracy and frequent resamplings, however this can be somewhat stemmed by using good resampling strategies and a more optimal importance function (further discussed by example in §3.1.2 and §3.1.3).

3.1.2 Resampling procedures

Figures 4(c) and 6 show that under multinomial resampling we have rapid degeneracy which as discussed in §2.5 is not surprising as it is poor in preserving the structure of the prior state estimate. In an analogous fashion to the *multinomial resampling* strategy of Figure 6, Figures 7, 8 and 9 show *systematic resampling*, *residual resampling* and *systematic within residual resampling* (residual resampling with a systematic resampling step instead of a multinomial resampling step for the remainder weightings) strategies respectively (as introduced in §2.5) applied to the same realization from the HMM as in §3.1.1 in order to find if they offer any substantive improvement.

Considering Figures 7, 8 and 9 we find that they all appear to perform better than multinomial resampling (being careful to draw conclusions from this as in each case we have but a single run of a stochastic algorithm). Referring to Figures 7(a), 8(a) and 9(a) note that in all alternate strategies the number of unique sample paths doesn't degenerate as rapidly (in effect a larger area under the black line), indicating they offer more diversity

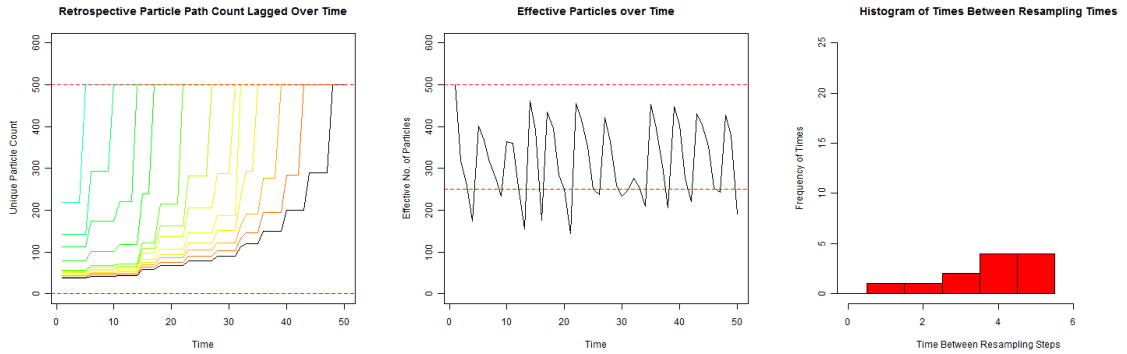


(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 6: Multinomial resampling prior importance function particle filter: Resampling diagnostic graphs.



(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 7: Systematic resampling prior importance function particle filter: Resampling diagnostic graphs.

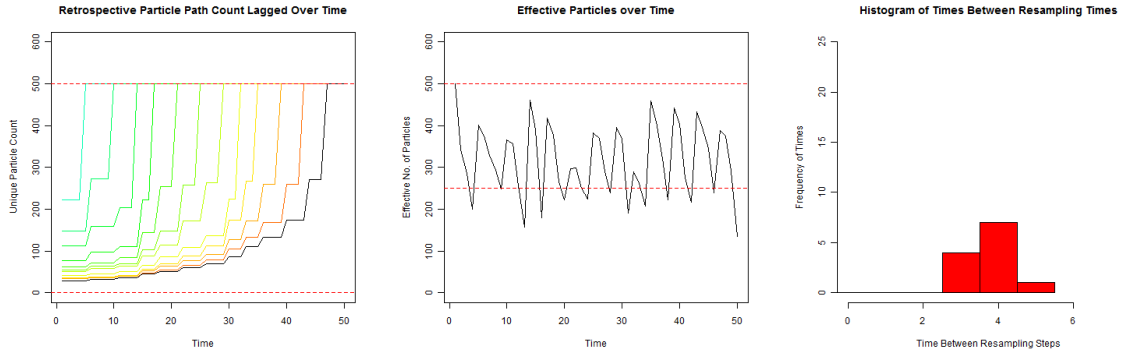
than in multinomial resampling. With further consideration of Figures 7(b), 8(b) and 9(b) and Figures 7(c), 8(c) and 9(c) we find that the *effective sample size* is more stable and the times of resampling are further apart than in the multinomial case. Comparison of Figures 7, 8 and 9 shows the differences between the strategies to be more minimal, with perhaps only some argument to be made that the *residual* and *systematic within residual* strategies have some advantage over the *systematic* strategy in that the times between resampling are greater.

To conclude, implementing any resampling procedure which preserves more of the structure than in the multinomial case can lead to significant gains in terms of variance reduction and computational cost (indeed, its difficult to find any reasonable procedure which preforms as poorly as it). Throughout the remainder of this report, unless otherwise stated a *systematic within residual* resampling strategy is employed.

3.1.3 Optimal importance function

A particle filter is applied using an *optimal importance function* (see §2.6.1) and a *systematic within residual* (see §2.5 and §3.1.2) resampling strategy to the simulation from the HMM used in §3.1.1 the results of which are shown in *Figure 10*. From Figure 10(a) we again appear to have a close fit to that of the Kalman filter, however comparing Figure 10(b) to Figure 4(b) we see the state estimate variance is now markedly better and centered around that of the Kalman filter.

Contrasting *Figure 10(c)* with Figure 4(c) notice that the number of completely unique particle paths is now much greater and they cover more of the tails of the prior state estimates. Looking at the diagnostic graphics in *Figure 11* and comparing them with the prior importance function in Figure 6 and the various resampling strategies

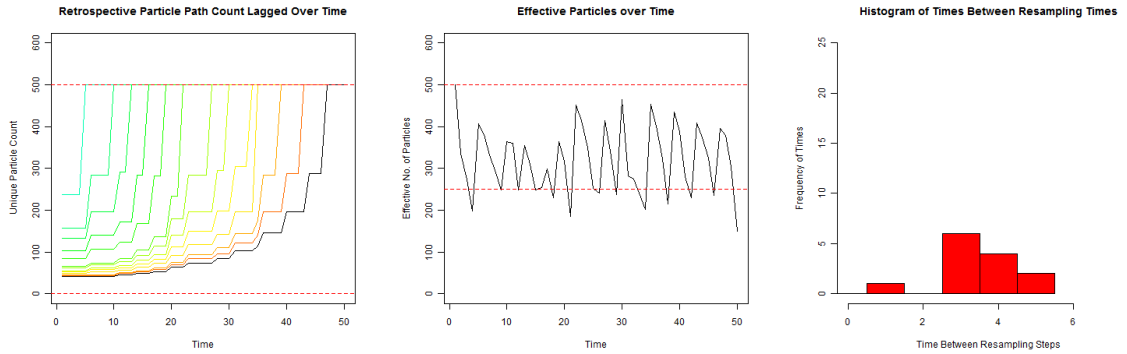


(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 8: Residual resampling prior importance function particle filter: Resampling diagnostic graphs.



(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 9: Systematic within residual resampling prior importance function particle filter: Resampling diagnostic graphs.

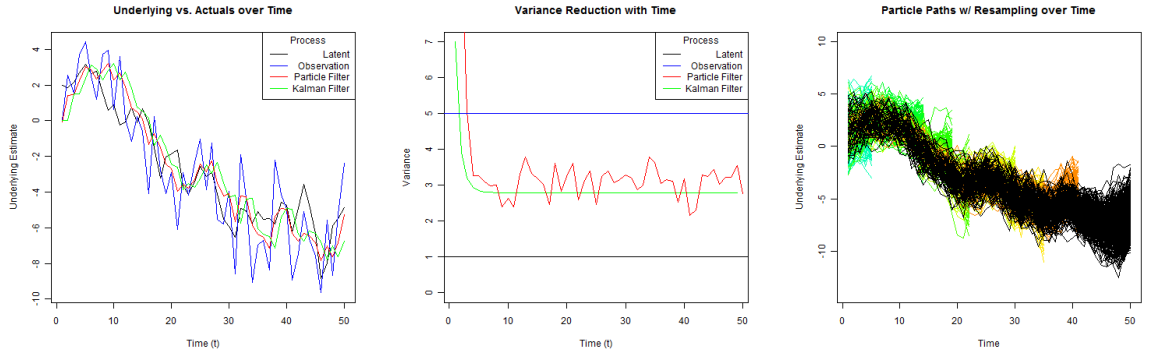
discussed in §3.1.2 we find that the optimal importance function performs much more favourably. In particular we now only resample seven times, *Figure 11(a)* shows lighter degeneracy with more than a tenth of the sample paths being entirely unique and *Figures 11(b) and 11(c)* show we now have far more stable effective sample size and greater times between resamplings.

Comparing *Figure 12(a)* to Figure 5(a) note that we now have more structure in the weightings of the particles which can be thought of as a result of fewer resamplings (as shown later) and comparing *Figure 12(c)* to Figure 5(c) notice that the estimated density is tighter with far lighter tails when using the optimal importance function.

3.2 Highly non-linear SSM

Consider the following (within the literature, canonical) non-linear HMM[9] applied to a realization of *50 time steps initialized at* $m_0 = 0.1$ with *initial density variance* $V_0 = 10$, *latent process variance* $V_t = 10\ \forall t \geq 1$ and

[9] In particular note that the observational process $(Y_t|X_t = x_t)$ is reflected on the x-axis, which results in interesting bi-modality in general (or in the unlikely case where the observation is negative uni-modality).

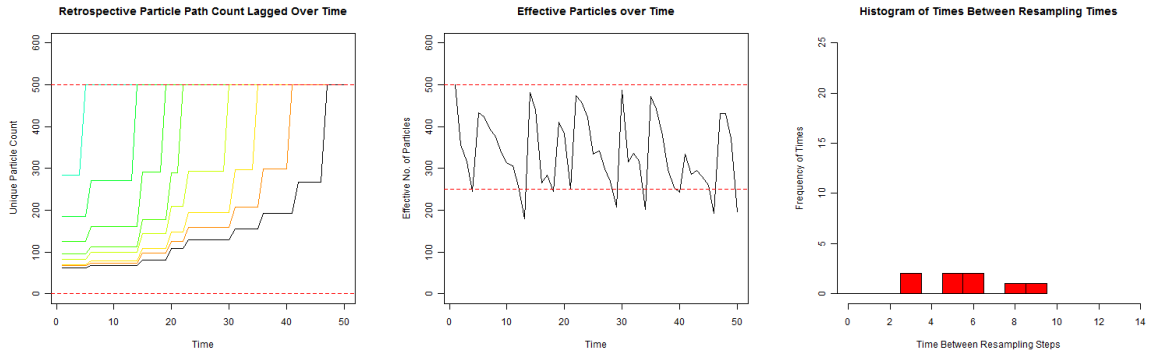


(a) Particle filter and Kalman filter estimate of latent process over time.

(b) Particle filter and Kalman filter state estimate variance over time.

(c) Heat map representing at each time point unique complete sample paths constituting the state estimate.

Figure 10: Optimal importance function particle filter applied to HMM with Gaussian SSDs.



(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 11: Optimal importance function particle filter: Resampling diagnostic graphs.

observational process variance $W_t = 5\ \forall t$ using a *particle set of size* 500,

$$X_1 \sim N\left(\frac{1}{2}m_0 + 25\frac{m_0}{1+m_0^2} + 8, \mathbf{C}_0\right) \tag{74}$$

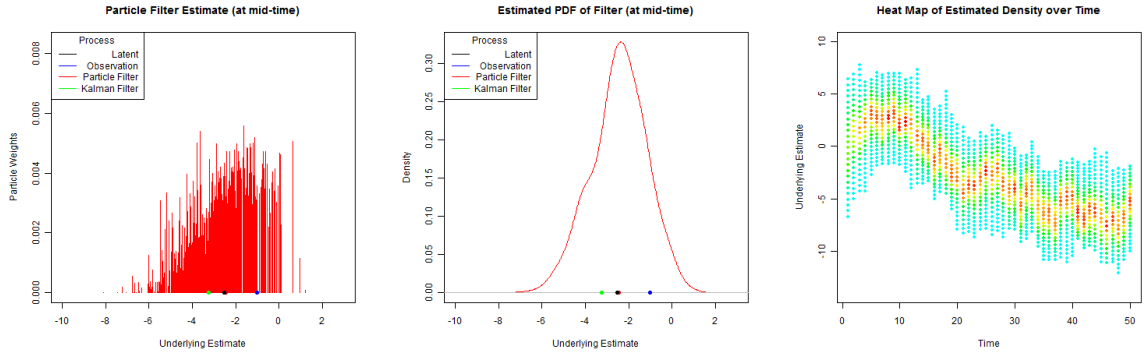
$$X_t|(X_{t-1} = x_{t-1}) \sim N\left(\frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos\left(\frac{6(t-1)}{5}\right), \mathbf{V}_t\right) \tag{75}$$

$$Y_t|(X_t = x_t) \sim N\left(\frac{x_t^2}{20}, \mathbf{W}_t\right) \tag{76}$$

3.2.1 Prior importance function

A particle filter with *prior importance function* and a *systematic within residual* resampling strategy is applied to a simulation from the HMM and corresponding parameterization of §3.2 and is shown in *Figure 13*. Referring to *Figure 13(a)* it is clear that the particle filter performs more poorly than before, in particular in the case where the latent process is close to zero or negative. *Figure 13(b)* illustrates frequent resampling, rapid unique path degeneracy and state estimates with large variance (which is further shown by the diagnostics of *Figure 14*).

From *Figure 15(a)* we find the importance weights are very smooth which is result of recent resampling. *Figure 15(c)* shows state estimates which are uninformative with very large variance.

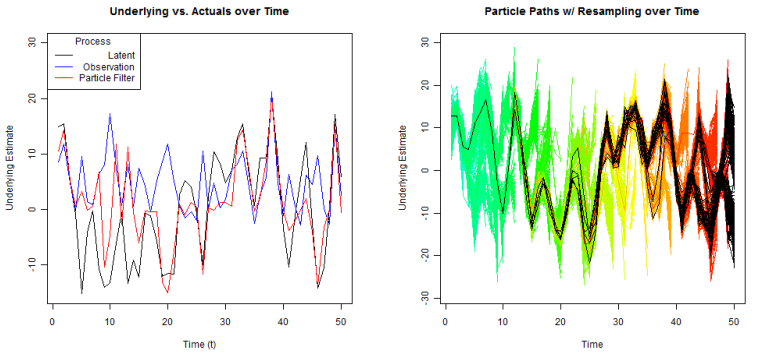


(a) Individual particles mid-time weightings.

(b) Mid-time empirical pdf evaluated using particle set.

(c) Empirical pdf evaluated using particle set at all times.

Figure 12: Optimal importance function filter particle set representation.



(a) Prior particle filter estimate of latent process over time.

(b) Heat map representing at each time point unique complete sample paths constituting the state estimate.

Figure 13: Prior importance function particle filter applied to a highly non-linear HMM.

3.2.2 Linearized optimal importance function

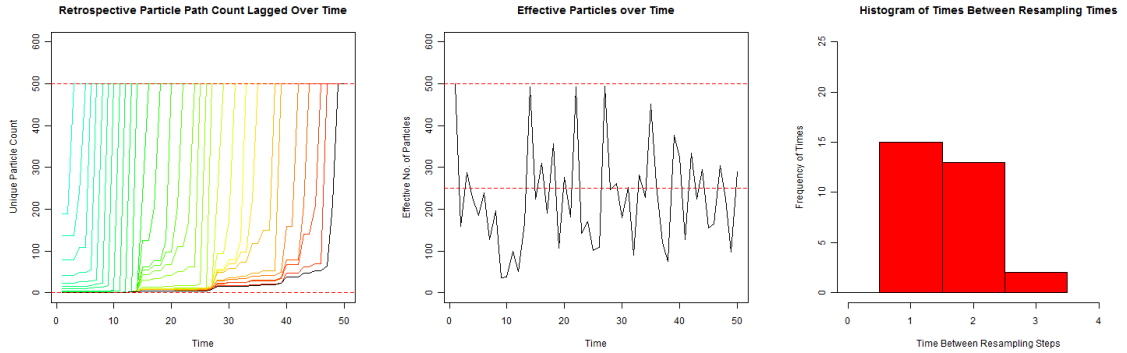
Despite using a strong resampling procedure the particle filter applied in §3.2.1 was ineffectual. Ideally, (as per §3.1.3) an *optimal importance function* would be used instead, however due to the non-linear nature of the HMM the dynamics are not sufficiently tractable to provide an exact optimal importance function and importance weights. Alternatively consider the *linearized optimal importance function* introduced in §2.6.1.2, in particular lets apply it to the same simulation from the HMM as in §3.2.1 and with a *systematic within residual* resampling procedure as before. For this model applying the results and notation of §2.6.1.2 we have,

$$f(x_{t-1}) = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos\left(\frac{6(t-1)}{5}\right) \qquad g(x_t) = \frac{x_t^2}{20} \tag{77}$$

$$M = \left.\frac{\partial g(x_t)}{\partial x_t}\right|_{x_t = f(x_{t-1})} = \frac{f(x_{t-1})}{10} \qquad D = -\frac{f(x_{t-1})^2}{10} \tag{78}$$

By applying directly these results to (61) to find the importance function and (64) for the importance weightings the *linearized optimal importance function* particle filter can be applied to the HMM, the results of which are shown in *Figure 16*.

Figure 16(a) appears to be a better fit than the prior importance function case of Figure 13(a), however there are still discrepancies in the estimate to the latent process particularly at the origin. The results of *Figure 16(b)* however are striking when compared to Figure 13(b) as we can immediately that the *linearized optimal impor-*

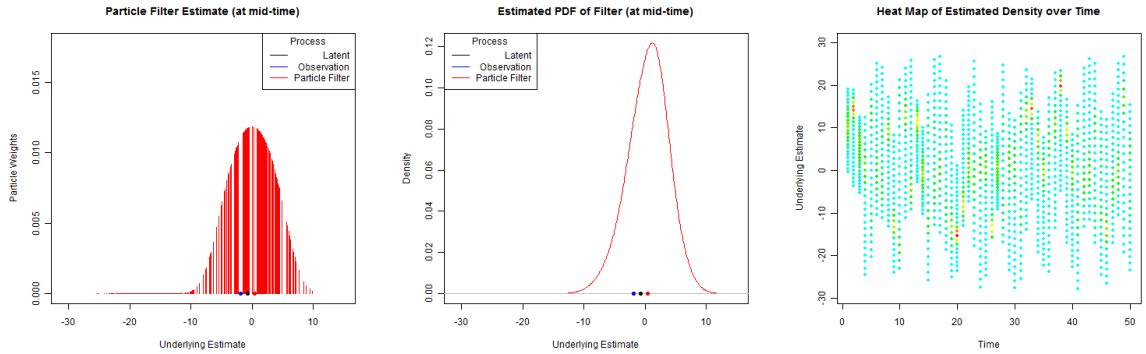


(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 14: Prior importance function particle filter: Resampling diagnostic graphs.



(a) Individual particles mid-time weightings.

(b) Mid-time empirical pdf evaluated using particle set.

(c) Empirical pdf evaluated using particle set at all times.

Figure 15: Prior importance function filter particle set representation.

tance function has the effect of reducing the number of resamplings and furthermore preserving a large number of completely unique particle sample paths for longer. This analysis is supported by *Figure 17(a)* which shows significantly lighter sample path degeneracy, *Figure 17(b)* showing a more stable *effective sample size* process and the fewer resamplings in *Figure 17(c)*. Further comparison of *Figure 18* to the prior case of Figure 15 shows cleaner state space estimates capturing the latent process and with lower state variance. To conclude, the *linearized optimal importance function* offers significant benefit over the prior importance function selection.

3.3 Bearings-only tracking

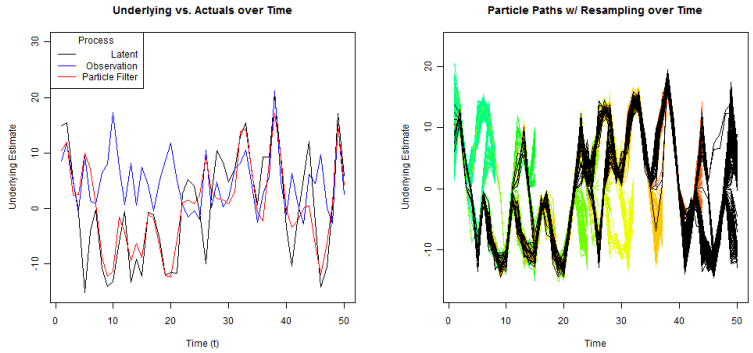
Consider a four dimensional target $x_t = (z_x, \bar{z_x}, z_y, \bar{z_y})^T$ which moves on the x-y plane (where (z_x, z_y) denotes the targets position and $(\bar{z_x}, \bar{z_y})$ denotes the targets velocity) according to the following HMM (an example originally introduced in [15][10]),

$$X_0 = x_0 \tag{79}$$

$$X_t|(X_{t-1} = x_{t-1}) \sim MVN(\mathbf{\Phi} x_{t-1}, \mathbf{\Gamma V}_t) \tag{80}$$

$$Y_t|(X_t = x_t) \sim N\left(\tan^{-1}\frac{z_y}{z_x}, W_t\right) \tag{81}$$

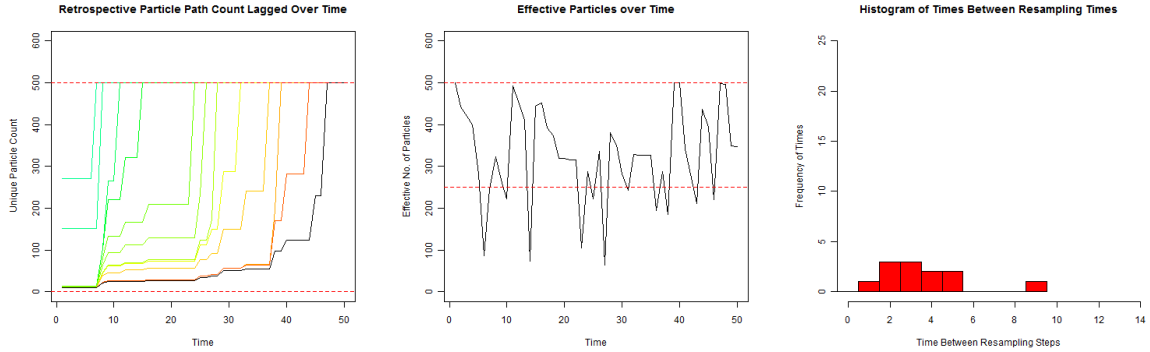
[10]This particular example can be nicely motivated by thinking of the toy example whereby a ship changes position and velocity over time with some random perturbation, while at the same time a person fixed at a particular point with some pre-notion of the ships location measures the ships bearings over time (again with some perturbation - perhaps instrument error), and attempts to infer from solely the time series of bearings the ships current position.



(a) Prior particle filter estimate of latent process over time.

(b) Heat map representing at each time point unique complete sample paths constituting the state estimate.

Figure 16: Linearized optimal importance function particle filter applied to a highly non-linear HMM.



(a) Unique sample path counts for each resampling backwards over time.

(b) Effective sample size over time.

(c) Histogram of time between resamplings.

Figure 17: Linearized optimal importance function particle filter: Resampling diagnostic graphs.

Furthermore, suppose we have some prior for the initial target position such that,

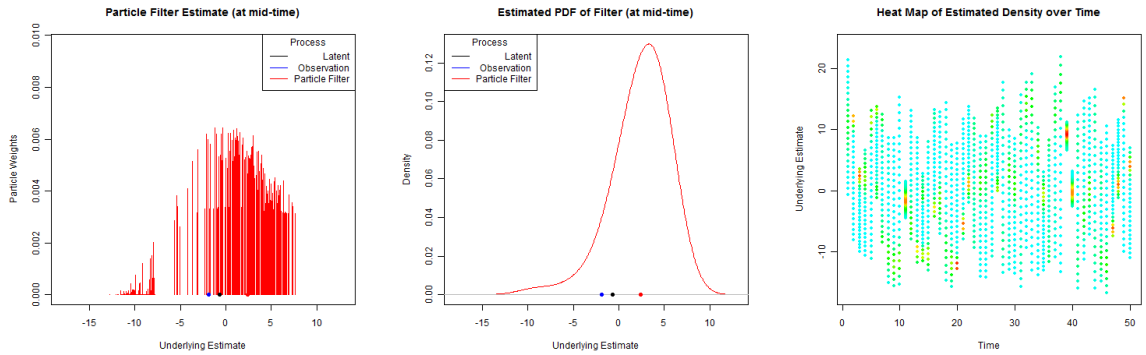
$$\tilde{X}_0 \sim MVN(m_0, \mathbf{M}_0) \tag{82}$$

For the purposes of this example it is assumed we are considering a single realization of *100 time steps* of this SSM with the following parameterization using a particle set of size 30,000,

$$x_0 = \begin{pmatrix} -0.05 \\ 0.001 \\ 0.7 \\ -0.055 \end{pmatrix} \quad \mathbf{\Phi} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{\Gamma} = \begin{pmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{pmatrix} \quad \mathbf{V}_t = \begin{pmatrix} (0.001)^2 \\ (0.001)^2 \end{pmatrix} \quad \mathbf{W}_t = (0.005)^2 \tag{83}$$

Furthermore, lets assume the parameterization of the prior to be,

$$m_0 = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.4 \\ -0.05 \end{pmatrix} \qquad \mathbf{M}_0 = \begin{pmatrix} (0.5)^2 & 0 & 0 & 0 \\ 0 & (0.005)^2 & 0 & 0 \\ 0 & 0 & (0.3)^2 & 0 \\ 0 & 0 & 0 & (0.01)^2 \end{pmatrix} \tag{84}$$

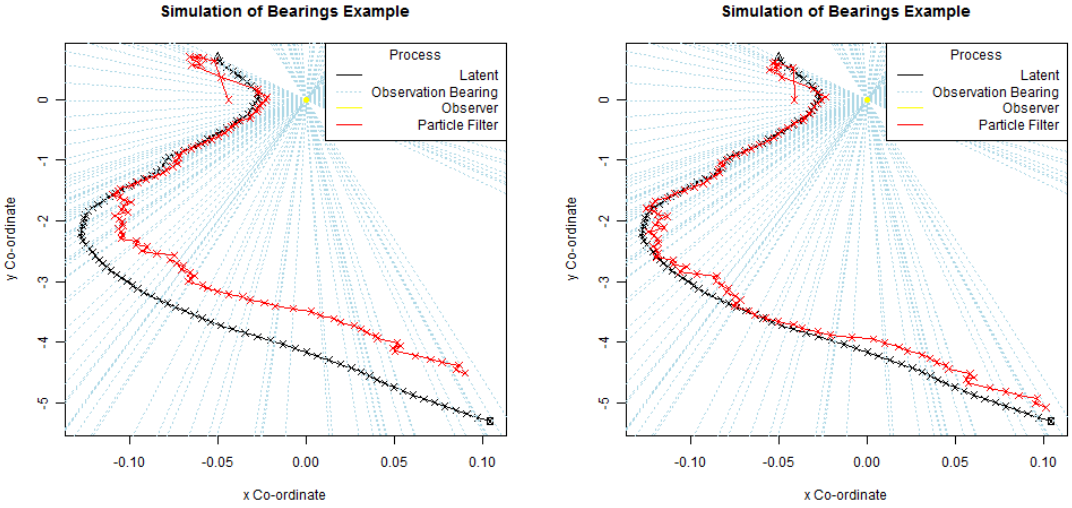


(a) Individual particles mid-time weightings. (b) Mid-time empirical pdf evaluated using particle set. (c) Empirical pdf evaluated using particle set at all times.

Figure 18: Linearized optimal importance function filter particle set representation.

3.3.1 Filter application

Figure 19 shows a single realization from the bearings-only tracking HMM (§3.3) overlayed with (in the case of *Figure 19(a)*) a *prior importance function* with *systematic within residual* resampling strategy particle filter, and (in the case of *Figure 19(b)*) an *auxiliary particle filter*. Comparison of the filters seems to suggest that the auxiliary particle filter performs extremely well and seems to capture the underlying latent process.



(a) Prior importance function.

(b) Auxiliary particle filter.

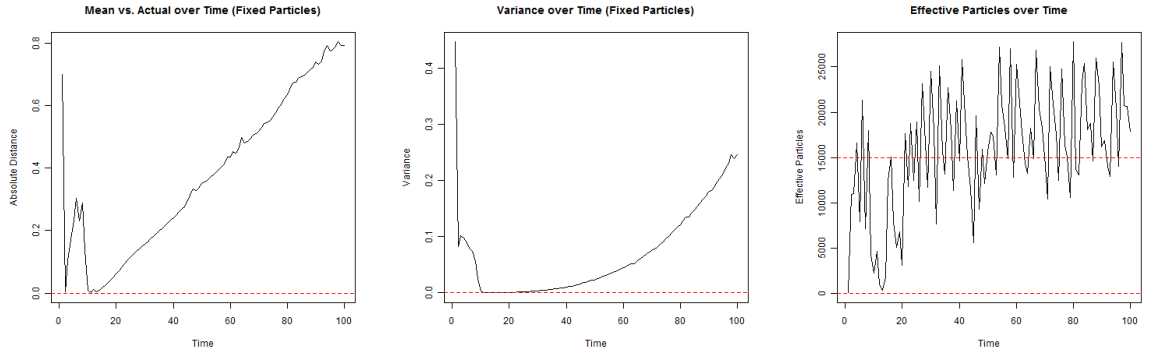
Figure 19: Bearings-only tracking filter application.

Unlike the 1-dimensional cases studied (see §3.1 and §3.2), reasonable graphical output and diagnostics are more difficult to find in this 4-dimensional case. Figure 19 does however anecdotally provide a great deal of intuition on the process. Notice that in this case the targets position initially drifts towards the bearings measurement point, before drifting away from the measurement point and finally panning across the x-y plane. As the target moves closer to the bearings measurement point it rapidly moves across bearings lines which can be thought of as the filter rapidly acquiring information on the latent state. Whenever the latent process begins moving on a near constant bearings line away from the origin the information received by the filter is negligible so the filter becomes poorer (with predilection towards the past) although along a bearings line being increasingly well estimated. As the target tracks across the x-y plane again more information is obtained though as the target is now far from the origin and moving away the information received is not sufficient to once again provide very good filter estimates.

In both the prior importance function case of *Figure 20* and the auxiliary particle filter case of *Figure 21* there is some indication that the anecdotal evidence holds. Considering *Figures 21(a) and 21(b)* compared to *Figures 20(a) and 20(b)* we see that although the auxiliary particle filter provides closer state estimates to the latent process in general they both display similar features. In particular, the initial state estimates are noisy due to the poor prior knowledge of the position of the target, however as the target moves towards the bearings measurement point the state estimates become very good, but subsequently it appears both filters are becoming *divergent* as the target drifts from the origin (see *Figures 20(b) and 21(b)*).

The effective sample size of the prior importance function shown in Figure 20(c) is highly erratic seemingly reflecting the nature of the HMM and the resampling procedure but note that there is some underlying structure to the process broadly in line with the anecdotal evidence. With comparison to *Figure 21(c)* we see that auxiliary particle filter has upon completing a burn-in a particularly stable effective sample size process indicating far better propagation of particles (also noting that there is some underlying structure as in the prior importance function case).

In conclusion in this particular HMM the *auxiliary particle filter* offers better estimates than the prior importance function but is still highly dependent on the path of the latent process and suffers from the lack of information available from the latent process when it is far from the origin.

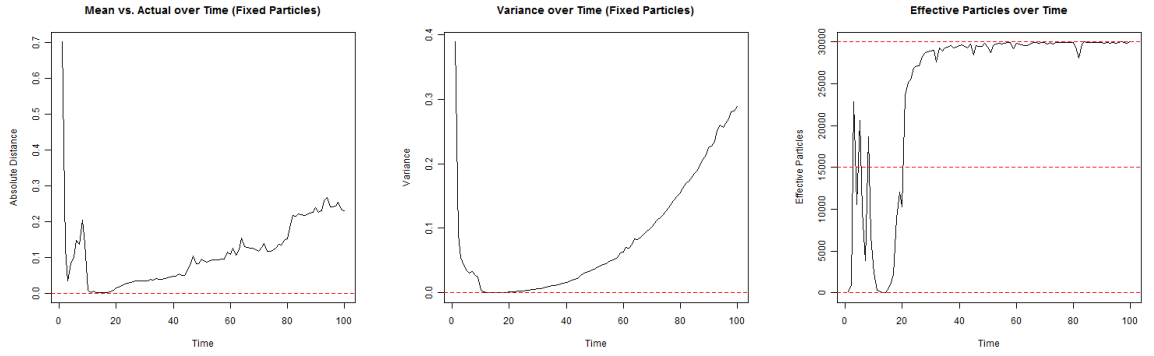


(a) Filter state estimate fit to latent process over time.

(b) Particle set variance over time.

(c) Effective sample size over time.

Figure 20: Prior importance function bearings-only tracking diagnostic graphics.



(a) Filter state estimate fit to latent process over time.

(b) Particle set variance over time.

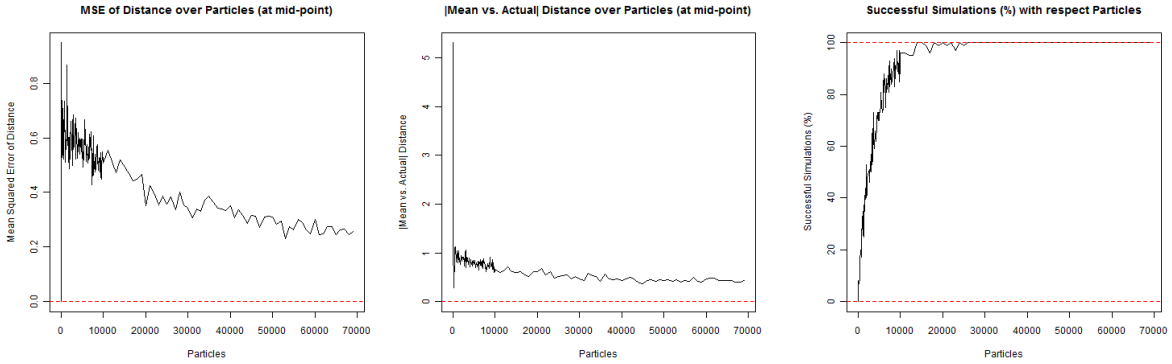
(c) Effective sample size over time.

Figure 21: Auxiliary particle filter bearings-only tracking diagnostic graphics.

3.3.2 A note on particle set size selection

The size of the particle set in all of the examples has been rather arbitrary reflecting the fact that there is no definitive methodology in order to make a good selection, however essentially there is a trade-off between increasing the number of particles to provide better state estimates with lower variance and increasing computational cost. To illustrate this trade-off the realization of the HMM in §3.3.1 is used to apply the prior importance function particle filter to particle sets of increasing size each 100 times the results of which for a single time point are shown in Figure 22(a). Here we find that initially there is considerable variance reduction from increasing the size of the particle set but this levels out for very large particle sets and as such it appears that 30,000 particles is of a reasonable size for this filter (however in reality the best solution would be to use any spare computational power to increase the size of the particle set).

Another consideration is that whenever the particle set is initialized and propagated in time reliable state estimates are only obtained if the particles are in the vicinity of the latent process, otherwise poorly placed individual particles begin to dominate the particle set after inevitable resamplings resulting in a filter which becomes steadily worse (*divergent*). Increasing the size of the particle set reduces the chances of such an event taking place. For our example simulation, *Figure 22(b)* illustrates the effect of increasing the size of the particle set and the corresponding reduction in divergence, again suggesting a set of 30,000 particles is of a reasonable size.



(a) Effect of an increase in the size of the particle set for a fixed time state estimate.

(b) Effect on the divergence rate of the particle filter for an increase in the size of the particle set.

Figure 22: Particle set size selection diagnostics.

References

[1] H. Akashi and H. Kumamoto. Random sampling approach to state estimation in switching environments. *Automatica.*, 13:429–434, 1977.

[2] B. Anderson and J. Moore. *Optimal filtering.* Prentice-Hall., 1st edition, 1979.

[3] L. Baum and J. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society.*, 73:360–363, 1967.

[4] L. Baum and T. Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Annals of Mathematical Statistics.*, 37(6):1554–1563, 1966.

[5] Y. Boers and J. Driessen. Multitarget particle filter track before detect application. *IEE Proceedings - Radar, Sonar and Navigation.*, 151(6):351–357, 2004.

[6] P.J. Brockwell and R.A. Davis. *Introduction to Time Series and Forecasting.* Springer., 2nd edition, 2002.

[7] J. Carpenter, P. Clifford, and P. Fearnhead. An Improved Particle Filter for Non-linear Problems. *IEE Proceedings - Radar, Sonar and Navigation.*, 146:2–7, 1999.

[8] J. Carpenter, P. Clifford, and P. Fearnhead. Building Robust Simulation-based Filters for Evolving Data Sets. Technical report, Department of Statistics, University of Oxford., 1999.

[9] N. Chopin. Central Limit Theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference. *The Annals of Statistics.*, 32(6):2385–2411, 2004.

[10] M.H.A. Davis and R.B. Vinter. *Stochastic Modelling and Control.* Chapman and Hall., 1st edition, 1985.

[11] J. F. G. De Freitas, M. A. Niranjan, A. H. Gee, and A. Doucet. Sequential Monte Carlo Methods to Train Neural Network Models. *Neural Computation.*, 12(4):955–993, 2000.

[12] A. Doucet, N. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice.* Springer., 1st edition, 2001.

[13] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing.*, 10:197–208, 2000.

[14] A. Doucet and A.M. Johansen. A Tutorial on Particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovsky, B., editor, *Oxford Handbook of Nonlinear Filtering.* Oxford University Press., 1st edition, To appear 2010.

[15] N. Gordon, J. Salmond, and A. Smith. A novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing.*, 140:107–113, 1993.

[16] J. E. Handschina and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *Journal of Control.*, 9:547–559, 1969.

[17] E.J. Hannan and M. Deistler. *The Statistical Theory of Linear Systems.* John Wiley., 1st edition, 1988.

[18] T. Higuchi. Monte carlo filter using the genetic algorithm operators. *Journal of Computational and Graphical Statistics.*, 59(1):1–23, 1997.

[19] G. Hovland and B. McCarragher. Hidden Markov Models as a Process Monitor in Robotic Assembly. *International Journal of Robotics Research*, 17(2):153–168, 1998.

[20] A.M. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics and Probability Letters.*, 78(12):1498–1504, 2008.

[21] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[22] R. Kalman and R. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering, Series D.*, 83:95–108, 1961.

[23] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME, Journal of Basic Engineering, Series D.*, 82:35–45, 1960.

[24] G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics.*, 5(1):1–25, 1996.

[25] A. Kong. A note on importance sampling using renormalized weights. Technical report, Department of Statistics, University of Chicago., 1992.

[26] A. Kong, S. Liu, and W. Wong. Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association.*, 89:278–288, 1994.

[27] A. Krogh, B. Larsson, G. von Heijne, and E. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *Journal of Molecular Biology.*, 305(3):567–580, 2001.

[28] J.S.. Liu and R. Chen. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association.*, 93(443):1032–1044, 1998.

[29] S. Mamon and R. Elliot. *Hidden Markov Models in Finance.* Springer., 1st edition, 2007.

[30] S. Maskell and N. Gordon. A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001.

[31] R. Meinhold and N. Singpurwalla. Understanding the Kalman Filter. *Journal of the American Statistical Association.*, 37(2):123–127, 1983.

[32] M. Pitt and N. Shepard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association.*, 94(446):590–599, 1999.

[33] M. Pitt and N. Shepard. Auxiliary Variable Particle Filters. In Doucet, A. and Freitas, N. and Gordon, N., editor, *Sequential Monte Carlo Methods in Practice.* Springer., 1st edition, 2001.

[34] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond The Annals of Statisticsthe Kalman filter: particle filters for tracking applications.* Artech House., 1st edition, 2004.

[35] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. In Bernardo, M. H. and Degroot, K. M. and Lindley, D. V. and Smith, A. F. M., editor, *Bayesian Statistics 3.* Oxford University Press., 1st edition, 1988.

[36] H. Tanizaki. Nonlinear Filters: Estimation and Applications. In *Lecture Notes in Economics and Mathematical Systems, Vol. 400.* Springer., 1993.

[37] H. Tanizaki and R.S Mariano. Prediction, filtering and smoothing in non-linear and non-normal cases using Monte Carlo integration. *Journal of Applied Econometrics.*, 9:163–179, 1994.

[38] P. Vidoni. Exponential family state space models based on a conjugate latent process. *Journal of the Royal Statistical Society, Series B (Statistical Methodology).*, 61(1):213–221, 1999.

[39] V.S. Zaritskii, V.B. Svetnik, and L.I. Shimelevich. Monte Carlo technique in problems of optimal data processing. *Automation and Remote Control.*, 12:95–103, 1975.