BAYESIAN STATISTICS 6, pp. 000--000
J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (Eds.)

Fixed-Lag Smoothing using Sequential Importance Sampling[1]

TIM C. CLAPP and SIMON J. GODSILL
Cambridge University Department of Engineering, UK

SUMMARY

In this paper we present methods for fixed-lag smoothing using Sequential Importance sampling (SIS) for state space models with unknown parameters. Sequential processing using Monte Carlo simulation is an area of growing interest for many engineering and statistical applications where data arrive point by point rather than in a batch. The methods presented here are related to the particle filtering ideas seen in Gordon *et al.* (1993), Liu and Chen (1995), Berzuini *et al.* (1997), Pitt and Shephard (1998) and Doucet *et al.* (1998). Techniques for fixed-lag simulation using either the filtering density or the smoothing density are developed. In addition we describe methods for regenerating parameters of the state-space model by sampling. We are concerned in particular with problems in Digital Communication systems where off-line or batch-based methods, such as Markov chain Monte Carlo (MCMC), are not well suited. The new techniques are demonstrated by application to a standard digital communications model and the performance of the various methods is compared.

Keywords: FIXED-LAG SMOOTHING, SAMPLING IMPORTANCE RESAMPLING (SIS), SEQUENTIAL MONTE CARLO, BLIND DECONVOLUTION, IMPORTANCE SAMPLING, PARTICLE FILTERS.

1. INTRODUCTION

The topic of Bayesian filtering using Monte Carlo techniques is a rapidly growing research area in many branches of science, statistics and engineering, since a sequential approach often provides a very natural way of processing large amounts of data that arrive point by point rather than in a batch. Monte Carlo methods have been shown to give dramatic improvements in performance for non-linear and non-Gaussian state-space models when compared with classical sequential techniques such as the extended Kalman filter (Anderson and Moore, 1979) and its variants.

The methods in this paper are developed from the particle filtering ideas presented in Gordon *et al.* (1993), Liu and Chen (1995), Berzuini *et al.* (1997) and Pitt and Shephard (1998). Particle filters work by representing the initial probability distribution by a set of points or *particles* drawn from that distribution. To update the samples from this probability distribution with the arrival of new data, weights are assigned to the particles according to their *importance* to the new distribution. If the variance of these weights becomes large, then only a small number of particles are contributing towards the representation of the probability distributions. We may *rejuvenate* by resampling the particles with probability according to their weights and reset weights to equal values. Clearly the probability distribution is only accurately described with a sufficiently high number of particles.

While the filtering problem is certainly the primary interest in many applications such as target tracking, in other applications such as digital communications or noise reduction for

[1]This work is supported by the Engineering and Physical Sciences Research Council.

speech signals, a small amount of delay, or latency, is allowable in the estimation of certain quantities. Then it will be profitable to perform fixed-lag smoothing rather than filtering (see e.g. Heller and Jacobs (1971) for a communications example). In this paper we address the problem of fixed-lag smoothing for state space models with fixed parameters. The methods developed are motivated by the structure of simple digital communications models which allow for very efficient simulation of the fixed-lag state vector.

We will assume a state-space form for the model of the data:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \boldsymbol{\theta}, \mathbf{b}_t)$$
$$\mathbf{y}_t = g(\mathbf{x}_t, \boldsymbol{\theta}, \mathbf{v}_t)$$

where $\mathbf{x}_t$ is the current state vector, $\mathbf{y}_t$ is the observation vector, $\boldsymbol{\theta}$ is a vector of unknown parameters and $\mathbf{v}_t$ is observation noise. In the communications model used later to illustrate the methods, the input sequence $\mathbf{b}_t$ is a discrete-valued sequence of transmitted symbols and hence $\mathbf{x}_t$ is a discrete Markov model. However, the techniques developed are also applicable when $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$ form a linear Gaussian state-space model conditional upon $\boldsymbol{\theta}$.

In the standard Monte Carlo filtering problem the aim is to simulate the state $\mathbf{x}_t$ from the filtering density, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, where $\mathbf{y}_{0:t}$ denotes the observation vectors from time 0 to t, inclusive. In the Monte Carlo fixed-lag problem we require a simulation from the smoothing density at lag p, that is $p(\mathbf{x}_{t-p}|\mathbf{y}_{0:t})$. For the model above there are two major technical challenges involved in this. The first is to handle the parameters $\boldsymbol{\theta}$, which are not straightforwardly incorporated into the basic frameworks of Gordon *et al.* (1993) or Pitt and Shephard (1998), since they require $\boldsymbol{\theta}$ to be dynamic rather than fixed. Berzuini *et al.* (1997), Liu and Chen (1995), and Liu and Chen (1998) give some insights into how the problem might be solved but do not give a complete solution for the general case. The second problem is the fixed-lag issue, which was also discussed by Pitt and Shephard in an early version of their 1998 paper. We provide some further insight into the solution of these challenging issues by considering state space models for which the following conditions apply:

- We can sample from $p(\boldsymbol{\theta}\,|\,\mathbf{x}, \mathbf{y})$, either directly or by some other method such as MCMC or rejection sampling.

- $\{\mathbf{x}_t\}$ is a discrete-valued Markov model and we can evaluate $p(\mathbf{y}\,|\,\mathbf{x}, \boldsymbol{\theta})$. Alternatively, $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$ form a linear Gaussian state-space model conditional upon $\boldsymbol{\theta}$.

Our methods are illustrated by an application from the field of digital communication systems. Each input data point is drawn from a finite set of symbols. We represent the transmission medium as a fixed filter with a finite impulse response (FIR), hence a discrete state-space system is formed. Conventional Markov chain Monte Carlo (MCMC) techniques such as the Gibbs sampler are unsuitable for this task because they can only perform processing on a batch of data. Data arrives sequentially, so it is advantageous to process it in this way. In addition, many communication systems are interactive, so there is a maximum level of latency that can be tolerated before a symbol is decoded. However, by allowing a delay of around 4$-$5 times the channel length, significant gains in performance can be obtained (Heller and Jacobs, 1971).

The paper is organised as follows: an introduction to importance sampling and the technique of sequential importance sampling is described in section 2. Section 3 describes two methods for achieving fixed-lag simulation using the filtering density—the differences arise from whether the parameters are marginalised analytically from the filtering density, the position of the filtering density tracked and the importance function used. Section 4 describes a method for fixed-lag simulation by updating the smoothing density. A comparison of the algorithms in the form of simulations on a digital communications system is presented in section 5.

2. SEQUENTIAL IMPORTANCE SAMPLING (SIS)

2.1. *Importance Sampling*

Suppose we are interested in the expectation of some function, $E\{f(\mathbf{x})\}$, but we cannot generate random draws from $p(\mathbf{x})$ directly. If $\pi(\mathbf{x})$ is a normalised density then we may write:

$$E\{f(\mathbf{x})\} = \int \frac{f(\mathbf{x})p(\mathbf{x})}{\pi(\mathbf{x})}\pi(\mathbf{x})\ dx$$

which may be estimated by drawing N samples, $\mathbf{x}^{(i)}$, from $\pi(\mathbf{x})$ and evaluating the expression:

$$E\{f(\mathbf{x})\} \approx \sum_{i=1}^{N} f(\mathbf{x}^{(i)}) \frac{w^{(i)}}{\sum_{j=1}^{N} w^{(j)}} \tag{1}$$

where $w^{(i)} = \frac{p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)})}$. The density $\pi(.)$ is called the importance function and the $w^{(i)}$ are the importance weights.

2.2. *Sequential Importance Sampling*

This section describes the basic technique of SIS in which the state distribution does not depend upon parameters, $\boldsymbol{\theta}$.

Suppose we have N samples, $\mathbf{x}_{0:t}^{(i)}$, from the probability distribution of interest at time t, $p(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t})$. With the arrival of a new data point $\mathbf{y}_{t+1}$, we would like to update this distribution to $p(\mathbf{x}_{0:t+1} \mid \mathbf{y}_{0:t+1})$ without modifying the simulated past trajectories, $\mathbf{x}_{0:t}^{(i)}$. Normally we can evaluate the likelihood $p(\mathbf{y}_t \mid \mathbf{x}_t)$ and the prior distribution $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$. We may update by using importance sampling sequentially, exploiting Bayesian updating at each time step as follows:

$$p(\mathbf{x}_{0:t+1} \mid \mathbf{y}_{0:t+1}) = \frac{p(\mathbf{y}_{t+1}, \mathbf{x}_{t+1} \mid \mathbf{x}_t)}{p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t})} \times p(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t}) \tag{2}$$

More generally, if an importance function of the form:

$$\begin{aligned}\pi(\mathbf{x}_{0:t+1} \mid \mathbf{y}_{0:t+1}) &= \pi(\mathbf{x}_{t+1} \mid \mathbf{x}_{0:t}, \mathbf{y}_{0:t+1})\, \pi(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t}) \\ &= \pi(\mathbf{x}_0 \mid \mathbf{y}_0) \prod_{k=1}^{t+1} \pi(\mathbf{x}_{k+1} \mid \mathbf{x}_{0:k}, \mathbf{y}_{0:k+1})\end{aligned}$$

is chosen, the importance weights may be evaluated recursively (Doucet *et al.* , 1998). Typically one cannot calculate the normalising term $p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t})$; however this is not required since it is independent of the state trajectory. The unnormalised incremental importance weights may then be calculated:

$$w_{t+1}^{(i)} = \frac{p(\mathbf{y}_{t+1}, \mathbf{x}_{t+1} \mid \mathbf{x}_t^{(i)})}{\pi(\mathbf{x}_{0:t+1} \mid \mathbf{x}_{0:t}^{(i)}, \mathbf{y}_{0:t+1})} \times w_t^{(i)}$$

In choosing a suitable importance function, any factorisation of the joint density, $p(\mathbf{y}_{t+1}, \mathbf{x}_{t+1} \mid \mathbf{x}_t)$, may be exploited so as to simplify this expression.

Statistics of interest may then be obtained from equation 1. If the variance of the weights becomes too large, we may resample the trajectories with the probability of each trajectory being selected equal to the corresponding weight. A useful measure of whether to resample is the effective sample size, described by Liu and Chen (1995), which tells us how many particles are actually contributing significantly to the distribution.

3. FIXED-LAG SIMULATION USING THE FILTERING DENSITY

The standard SIS technique may be adapted to achieve fixed-lag smoothing, whilst updating the filtering density. We consider two ways of achieving this: standard SIS (with parameters marginalised) in which the state history is stored back to a lag of p, and a new technique in which the states and parameters are both drawn sequentially.

3.1. *SIS and Sequential Imputations*

The simplest method for fixed-lag simulation is to update the filtering density as each data point arrives as in standard SIS, i.e. track the evolution of $p(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t})$, storing the state history of each particle back to a lag of p states. For this type of updating we may use any of the standard importance distributions ranging from the prior distribution, $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$, as described by Handschin and Mayne (1969) and Gordon *et al.* (1993) to the optimal importance function, $p(\mathbf{x}_{t+1} \mid \mathbf{y}_{t+1}, \mathbf{x}_t)$, described by Zaritskii *et al.* (1975) and Liu and Chen (1995) (under the name of sequential imputations). For a review of these methods see Doucet *et al.* (1998). This direct approach requires analytic marginalisation of the parameters $\boldsymbol{\theta}$, which is in fact possible for the communications model we shall consider, but will not be possible for many other models of practical interest.

The optimal method for filtering the data without lookahead is the method of *sequential imputations* as described by Liu and Chen (1995) (see Doucet *et al.* (1998) for a discussion of this). At each step only a draw of the state from $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{y}_{t+1})$ is required. The incremental importance weight is given by $p(\mathbf{y}_{t+1} \mid \mathbf{x}_t)$, and updates of the mean and variance of the marginal distribution of the parameters, $\boldsymbol{\theta}$, are obtained by recursive relationships derived from the matrix inversion lemma in the conditionally Gaussian parameter case.

Given the stored histories of each particle back to time $t-p$ it is then straightforward, using the importance weights from time t, to make an estimate of the smoothing density, from which estimates such as the MAP estimate may be extracted (we refer to this method as 'sequential imputations with decision step'). The performance of the fixed-lag MAP approach is compared in our simulations with the standard sequential imputations methods in which the joint MAP state sequence is estimated after the arrival of all the data.

3.2. *Lagged time filtering density*

As an alternative, we may attempt to draw the fixed-lag state and the parameter jointly, a procedure with more generality than sequential imputations, since it will not require a marginalisation of the parameters. To simplify notation, we will shift our time-base forward by p samples. Hence at time t we will draw fixed-lag samples of $\mathbf{x}_t$ using information from $\mathbf{y}_{0:t+p}$.

The joint prediction density may be factorised as follows:

$$p(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}, \boldsymbol{\theta} \mid \mathbf{x}_{0:t}, \mathbf{y}_{0:t}) = p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta}, \mathbf{y}_{t+1})\, p(\mathbf{y}_{t+1} \mid \mathbf{x}_t, \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{x}_{0:t}, \mathbf{y}_{0:t})$$

By using this factorisation, new draws for the parameters can be made at each time without having to draw the states from the prior density, $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{y}_t)$, as suggested by Liu and Chen

(1998). It is possible to sample directly from the first term; however we have chosen instead to use the importance function:

$$\pi(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta}, \mathbf{y}_{t+1:t+p+1}) = p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta}, \mathbf{y}_{t+1:t+p+1}) \tag{3}$$

with the hope of using the additional data to bias the draw towards states that will be useful at later iterations. This may be accomplished by using forward filtering backward sampling techniques described by Carter and Kohn (1994), Doucet and Duvaut (1996) and Clapp and Godsill (1997), drawing from $p(\mathbf{x}_{t+1:t+p+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta}, \mathbf{y}_{t+1:t+p+1})$ and discarding the unwanted imputed values $\mathbf{x}_{t+2:t+p+1}$. This introduces the importance weight:

$$\begin{aligned} w_{t+1} &= \frac{p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta}, \mathbf{y}_{t+1})}{\pi(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta}, \mathbf{y}_{t+1:t+p+1})}\, w_t && (4)\\ &= \frac{p(\mathbf{y}_{t+1} \,|\, \mathbf{x}_{t+1}, \boldsymbol{\theta})\, p(\mathbf{y}_{t+1:t+p+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta})}{p(\mathbf{y}_{t+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta})\, p(\mathbf{y}_{t+1:t+p+1} \,|\, \mathbf{x}_{t+1}, \boldsymbol{\theta})}\, w_t && (5) \end{aligned}$$

The first term of both numerator and denominator are easy to compute. For calculation of the second terms, note the following factorisation:

$$\begin{aligned} p(\mathbf{y}_{t:t-p+1} \,|\, \boldsymbol{\theta}, \mathbf{x}_{t-p}) = p(\mathbf{y}_t \,|\, \boldsymbol{\theta}, \mathbf{y}_{t-1:t-p+1}, \mathbf{x}_{t-p}) \;\times& \\ p(\mathbf{y}_{t-1} \,|\, \boldsymbol{\theta}, \mathbf{y}_{t-2:t-p+1}, \mathbf{x}_{t-p}) \ldots p(\mathbf{y}_{t-p+1} \,|\, \boldsymbol{\theta}, \mathbf{x}_{t-p})& \end{aligned} \tag{6}$$

Each of these terms may be expressed as:

$$p(\mathbf{y}_{t-k} \,|\, \boldsymbol{\theta}, \mathbf{y}_{t-k-1:t-p+1}, \mathbf{x}_{t-p}) = \sum_{\mathbf{x}_{t-k}} p(\mathbf{y}_{t-k} \,|\, \boldsymbol{\theta}, \mathbf{x}_{t-k}, \mathbf{x}_{t-p})\, p(\mathbf{x}_{t-k} \,|\, \boldsymbol{\theta}, y_{t-p+1:t-k-1}, \mathbf{x}_{t-p}) \tag{7}$$

The first term is the likelihood and the second is the prediction density, both of which are calculated during the forward pass when making the draw from equation 3. Although a separate forward pass is needed for the calculation of $p(\mathbf{y}_{t+1:t+p+1} \,|\, \mathbf{x}_{t+1}, \boldsymbol{\theta})$, this is not required when calculating $p(\mathbf{y}_{t+1:t+p+1} \,|\, \mathbf{x}_t, \boldsymbol{\theta})$.

Note that in this case the draw for the parameters does not include any information from the extra received data, $\mathbf{y}_{t+1:t+p+1}$. However, in the case of fixed parameters, the prior rapidly becomes quite strong so any additional information in these states does not affect the posterior for $\boldsymbol{\theta}$ very much.

3.2.1. *Method*

Let us assume we have available at time t, N samples, $\mathbf{x}_t^{(j)}$, from the distribution $p(\mathbf{x}_t \,|\, \mathbf{y}_{0:t})$,[2] possibly with associated weights $w_t^{(j)}$. At time $t+1$ the additional observation $\mathbf{y}_{t+p+1}$ is available. These samples may be updated to the distribution $p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{0:t+1})$ as follows:

1. For each j:

   (a) Draw $\boldsymbol{\theta}^{(j)}$ from $p(\boldsymbol{\theta} \,|\, \mathbf{x}_{0:t}^{(j)}, \mathbf{y}_{0:t})$.

   (b) Draw $\mathbf{x}_{t+1}^{(j)}$ from equation 3.

[2] Initial samples could be created by any batch-based MCMC method.

(c) Calculate importance weights from equation 5.

2. If weights have a high variance (or equivalently, the effective sample size is too low), rejuvenate by resampling with probability proportional to the weights and reset the weights to equal values.

4. FIXED-LAG SIMULATION USING THE SMOOTHING DENSITY

Rather than using the filtering density, $p(\mathbf{x}_{0:t} \mid \mathbf{y}_{0:t})$, the smoothing density, $p(\mathbf{x}_{0:t-p} \mid \mathbf{y}_{0:t})$, may be used to achieve fixed-lag smoothing. Ideologically this is a better approach, although it introduces some practical problems.

The fixed-lag smoothing distribution is given by:

$$\begin{aligned} p(\mathbf{x}_{0:t-p+1} \mid \mathbf{y}_{0:t+1}) &= \frac{p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p+1})\, p(\mathbf{x}_{0:t-p+1} \mid \mathbf{y}_{0:t})}{p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t})} \\ &= \frac{p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t}, \mathbf{x}_{t-p+1})\, p(\mathbf{x}_{t-p+1} \mid \mathbf{y}_{t-p+1:t}, \mathbf{x}_{0:t-p})\, p(\mathbf{x}_{0:t-p} \mid \mathbf{y}_{0:t})}{p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t})} \end{aligned} \tag{8}$$

Let us define the importance sampling distribution in the usual way:

$$\begin{aligned} \pi(\mathbf{x}_{0:t-p+1} \mid \mathbf{y}_{0:t+1}) &= \pi(\mathbf{x}_0 \mid \mathbf{y}_{0:p}) \prod_{k=1}^{t-p+1} \pi\left(\mathbf{x}_k \mid \mathbf{x}_{0:k-1,}\mathbf{y}_{0:k+p}\right) \\ &= \pi(\mathbf{x}_{t-p+1} \mid \mathbf{x}_{0:t-p}, \mathbf{y}_{0:t+1})\, \pi(\mathbf{x}_{0:t-p} \mid \mathbf{y}_{0:t}) \end{aligned}$$

then the unnormalised importance weight satisfies:

$$\begin{aligned} w(\mathbf{x}_{0:t+1}) &= \frac{p(\mathbf{x}_{0:t-p+1} \mid \mathbf{y}_{0:t+1})}{\pi(\mathbf{x}_{0:t-p+1} \mid \mathbf{y}_{0:t+1})} \\ &= w(\mathbf{x}_{0:t})\, \frac{p(\mathbf{y}_{t+1} \mid \mathbf{y}_{t-p:t}, \mathbf{x}_{0:t-p+1})\, p(\mathbf{x}_{t-p+1} \mid \mathbf{y}_{t-p+1:t}, \mathbf{x}_{0:t-p})}{\pi(\mathbf{x}_{t-p+1} \mid \mathbf{x}_{0:t-p}, \mathbf{y}_{0:t+1})\, p(\mathbf{y}_{t+1} \mid \mathbf{y}_{0:t})} \end{aligned} \tag{9}$$

For a general distribution, the term $p(\mathbf{y}_{t+1} \mid \mathbf{y}_{t-p:t}, \mathbf{x}_{0:t-p+1})$ cannot be evaluated. In some models we may assume that for a sufficiently large value of the lag, p, the term $p(\mathbf{y}_{t+1} \mid \mathbf{y}_{t-p:t}, \mathbf{x}_{0:t-p+1})$ will be approximately constant for all values of $\mathbf{x}_{0:t-p+1}$.[3] This is saying simply that future observations are independent of states from the distant past.

We now consider generating samples from the joint distribution $p(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y})$ by using the following factorisation:

$$p(\mathbf{x}_{t-p+1}, \boldsymbol{\theta} \mid \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}) = p(\mathbf{x}_{t-p+1} \mid \mathbf{x}_{0:t-p}, \mathbf{y}_{t-p+1:t}, \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}) \tag{10}$$

Note that we can easily sample from the distribution:

$$p(\mathbf{x}_{t-p+1}, \mathbf{x}_{t-p+2:t} \mid \mathbf{x}_{0:t-p}, \mathbf{y}_{t-p+1:t}, \boldsymbol{\theta}) \tag{11}$$

[3] A similar approximation is made in the communications literature concerning trace-back in a Viterbi equaliser Proakis (1995). After a fixed delay the paths stored in the Viterbi algorithm are assumed to have joined, or if not the most likely path is chosen at this time.

as before. The imputed values, $\widetilde{\mathbf{x}}_{t-p+2:t}$, are retained for use in sampling $\boldsymbol{\theta}$ in the next iteration. We cannot easily sample from $p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p})$; however we may sample from the importance distribution:

$$\pi(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}) = p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}, \widetilde{\mathbf{x}}_{t-p+1:t}) \tag{12}$$

This introduces an incremental importance weight:

$$\begin{aligned} w_{t+1} &= \frac{p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p})}{p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}, \widetilde{\mathbf{x}}_{t-p+1:t})}\, w_t \\ &= \frac{\sum_{\widetilde{\mathbf{x}}_{t-p+1:t}} p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}, \widetilde{\mathbf{x}}_{t-p+1:t})}{p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:t}, \mathbf{x}_{0:t-p}, \widetilde{\mathbf{x}}_{t-p+1:t})}\, w_t \end{aligned} \tag{13}$$

At first glance it may appear that the numerator of equation 13 is expensive to calculate, however all the terms will be calculated when sampling from equation 11; only a small number of additional calculations are required.

4.1. *Method*

Let us assume we have available at time t, N samples, $\mathbf{x}_{t-p}^{(j)}$, from the distribution $p(\mathbf{x}_{t-p}^{(j)} \,|\, \mathbf{y}_{0:t})$, possibly with associated weights $w^{(j)}$. At time $t+1$ the additional observation $\mathbf{y}_{t+1}$ is available. These samples may be updated to the distribution $p(\mathbf{x}_{t-p+1} \,|\, \mathbf{y}_{0:t+1})$ as follows:

1. For each j:
   (a) Draw $\boldsymbol{\theta}^{(j)}$ from equation 12.
   (b) Draw $\mathbf{x}_{t-p+1}^{(j)}$, $\widetilde{\mathbf{x}}_{t-p+2:t}^{(j)}$ from equation 11 and draw the additional $\widetilde{\mathbf{x}}_{t+1}^{(j)}$ from $p(\mathbf{x}_{t+1} | \mathbf{y}_{t+1}, \widetilde{\mathbf{x}}_{t}^{(j)}, \boldsymbol{\theta})$.
   (c) Calculate importance weights from equation 13.
2. If weights have a high variance (or equivalently, the effective sample size is too low), rejuvenate by resampling with probability proportional to the weights and reset the weights to equal values.

5. SIMULATIONS

5.1. *Problem Formulation*

Most digital communications systems transmit a signal $\{b_t\}$ where the value of each of the b_t are taken from a finite alphabet of q symbols. This is transmitted over a channel which may introduce distortion and noise. The channel model used here is an FIR filter with additive Gaussian noise:

$$y_t = \sum_{i=0}^{n-1} b_{t-i} h_i + v_t \tag{14}$$

$$b_t \in \{S_0, \ldots, S_{q-1}\} \tag{15}$$

$$v_t \overset{i.i.d.}{\sim} N(0, \sigma_v^2) \tag{16}$$

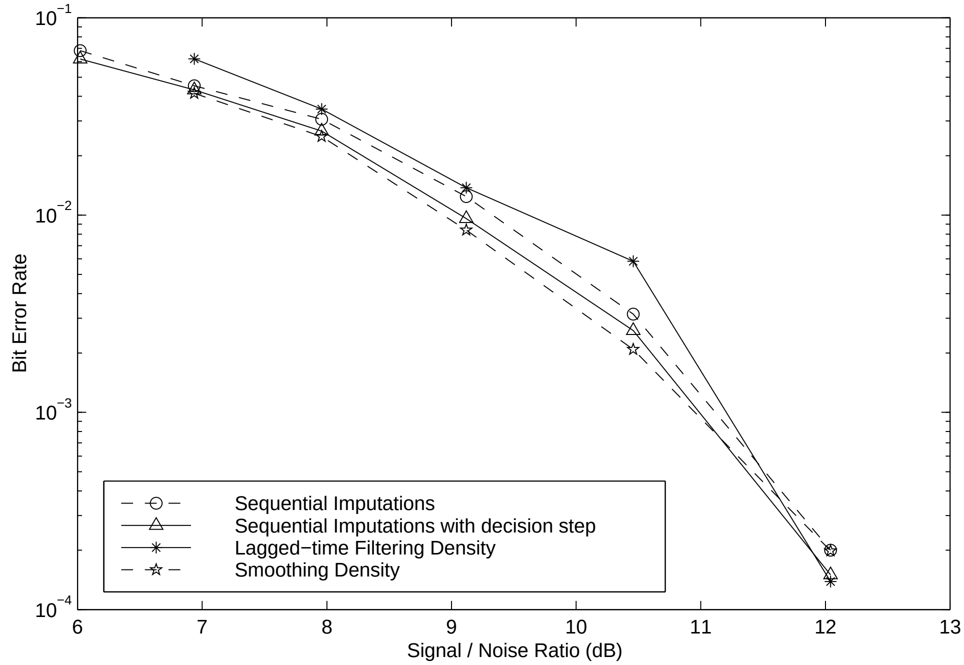

Figure 1. *Graph of Bit Error Rate against Signal-to-Noise ratio in simulations using the different SIS algorithms.*

where $\{y_t\}$ is the observed signal, $\{b_t\}$ is the transmitted signal, S_i a symbol from the alphabet, $\mathbf{h} = \{h_0, \ldots, h_{n-1}\}$ is the channel filter, n is the number of filter taps and t is an integer.

This may be represented in an equivalent state-space form:

$$\begin{cases} \mathbf{x}_t &= \left[\begin{array}{c|c} \mathbf{0}^{\mathbf{T}} & 0 \\ \hline \mathrm{I} & \mathbf{0} \end{array}\right] \mathbf{x}_{t-1} + \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} b_t \\ y_t &= \mathbf{h}^{\mathbf{T}} \mathbf{x}_t + v_t \end{cases} \tag{17}$$

where $\mathbf{x}_t = [b_t, b_{t-1}, \ldots, b_{t-n+1}]^{\mathbf{T}}$ is the state (a vector of length n) and $\mathbf{0}$ is the null vector of length $n-1$. A collection of states, $\mathbf{x}_{1:t}$, can be exactly determined by $\{b_1, \ldots, b_t\}$ and some initial conditions, given equation 17.

5.2. *Results*

To compare methods we tried them on communications channels simulated directly from equation 14 using a binary amplitude modulation scheme where bits are encoded as ± 1. All methods require an estimate of the initial density: this was obtained using a Gibbs sampler (see Clapp and Godsill (1997) for details) which was run over a frame of length 200 samples. The channel was chosen to be $\mathbf{h} = [0.766,\ 0.575,\ 0.287]^{\mathbf{T}}$ and the lag $p = 15$, five times the channel length. In all cases 151 particle streams were chosen. This is an order of magnitude lower than many implementations of the particle filter, however it is sufficient for our purposes since we can produce a large number of points close to the posterior modes due to the efficient algorithms employed and the comparative simplicity of the discrete state-space system. For the case of the current time filtering density (sequential imputations) algorithm and the update of the smoothing density, rejuvenation was carried out when the effective sample size fell below 135. The method using the lagged time filtering density produced a much wider variance in the weights, so resampling took place at each step. The input bit at time t was determined by summing over all states at time t which would have that bit set (this corresponds to a marginal MAP estimate).

The performance of the various algorithms are shown in figure 1. Differential encoding was not employed, but the bits were inverted when the phase ambiguity was incorrectly resolved (choosing $\mathbf{h} = [-0.766, -0.575, -0.287]^{\mathbf{T}}$ is an equally probable solution). In a practical system, either differential encoding or a small number of known bits are required. Note that the error bounds on the points at very high signal to noise ratios are quite large due to a very low number of errors being detected. This could account for the curves swapping over.

5.3. *Comparison of Algorithms*

All algorithms performed well. For comparison the performance of a Viterbi equaliser using 20 known symbols in a frame of 200, the channel estimate being obtained by the least mean squared error, lies somewhere between the dashed circle and the solid triangle curves. The Viterbi algorithm performs maximum likelihood sequence estimation given the channel parameters and is commonly used in practical applications e.g. GSM/ETSI mobile phones. Errors typically occurred in bursts: the burst could last up to 4 symbols (not necessarily all would be in error), though 2 was most common. Rejuvenation occurred most iterations on the sequential imputations algorithm, although the gap increased to about every 10 iterations with less noise present. When using the smoothing density, typical values for the time gap between rejuvenation varied between 150 and 700 as noise levels decreased.

The comparatively poor performance of the lagged-time filtering density algorithm is perhaps due to this algorithm essentially behaving as a directed filter rather than a smoothing algorithm. All of our other algorithms give a genuinely smoothed output; however it is trivial to add a delayed decision step in the same way as for SIS.

It is interesting to note that using the sequential imputations algorithm with the decision step actually performs better than retaining all the past states data until the end and then making a decision. The likely reason for this is because the states drawn are not necessarily good ones and the frequent rejuvenation is required. This repeated resampling leads to degeneracy in the earlier states.

Computationally the lagged draw of a state (or equivalently a joint draw of the states) is the most expensive part of the algorithms. As a result sequential imputations is by far the fastest method, and using the lagged-time filtering density was slowest (by almost a factor of 2) due to the extra pass through the lagged data required for calculation of the importance weights.

6. CONCLUSION

In this paper we have explored some of the issues involved in performing sequential simulations for state space models with unknown parameters, presenting algorithms for filtering and fixed-lag smoothing which can be applied to many of the models current in engineering time series. The utility of the algorithms has been demonstrated by application to a basic digital communications model. There are as yet, however, several unresolved technical issues in this challenging area, and we hope the methods presented here will give some further insight into solving the sequential simulation problem in all its generality.

REFERENCES

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall.

Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997). Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Amer. Statist. Assoc.* **92**(440), 1403--1412.

Carter, C. K. and Kohn, R. (1994). Gibbs sampling for state space models. *Biometrika* **81**(3), 541--553.

Clapp, T. and Godsill, S. J. (1997). Bayesian Blind Deconvolution for Mobile Communications. *Proceedings of IEE Colloqium on Adaptive Signal Processing for Mobile Communication Systems*, 9/1--9/6. Savoy Place, London. Reference No: 1997/383.

Doucet, A. and Godsill (1998). On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. Submitted for publication. Available as Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering.

Doucet, A. and Duvaut, P. (1996). Fully Bayesian Analysis of Hidden Markov models. *Proceedings of EUSIPCO*.

Gelman, A. and Carlin, J. B. and Stern, H. S. and Rubin, D. B. (1995). *Bayesian Data Analysis*. London: Chapman and Hall.

Gordon, N., Salmond, D. and Ewing, C. (1995). Bayesian State Estimation for Tracking and Guidance using the Bootstrap Filter. *Journal of Guidance Control and Dynamics* **18**(6), 1434--1443.

Gordon, N., Salmond, D. and Smith, A. F. M. (1993). Non-linear / Non-Gaussian Bayesian State Estimation. *IEE Proceedings of Radar and Signal Processing, Pt.F*, **140**(2), 107--113.

Handschin, J. and Mayne, D. (1969). Monte Carlo Techniques to Estimate the Conditional Expectation in Multi-stage Non-Linear Filtering. *Int. J. Cont.*, volume 9, 547--559.

Heller, J. A. and Jacobs, I. M. (1971). Viterbi Decoding for Satellite and Space Communication. *IEEE Trans. Communcation Technology* **COM-19**, 835--848.

Kong, A., Liu, J. and Wong, W. H. (1994). Sequential Imputations and Bayesian missing data problems. *J. Amer. Statist. Assoc.* **89**(425), 278--288.

Liu, J. and Chen, R. (1995). Blind Deconvolution via Sequential Imputations. *J. Amer. Statist. Assoc.* **90**(430), 567--576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamical systems, *J. Amer. Statist. Assoc.* **93**.

Pitt, M. and Shephard, N. (1998). Filtering via Simulation: Auxiliary Particle Filters, *J. Amer. Statist. Assoc.* (to appear).

Proakis, J. G. (1995). *Digital Communications*. Electrical engineering series. New York: McGraw-Hill, 3rd edition.

Ross, S. M. (1997) *Simulation*. Statistical modelling and decision science series. New York: Academic Press, 2nd edition.

Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian Statistics without Tears: A Sampling-Resampling Perspective. *Amer. Statist.* **46**(2), 84--88.

Zaritskii, V., Svetnik, V. and Shimelevich, L. (1975). Monte Carlo Technique in Problems of Optimal Data Processing. *Auto. Remo. Cont.* **12**, 95--103.