Building Robust Simulation-based Filters for Evolving Data Sets

James Carpenter, Peter Clifford† and Paul Fearnhead

† *Address for correspondence:* Department of Statistics, University of Oxford, 1 South Parks Road, Oxford, OX1 3TG, UK. Email: *clifford@stats.ox.ac.uk*

Summary

The need for accurate monitoring and analysis of sequential data arises in many scientific, industrial and financial problems. Although the Kalman filter is effective in the linear-Gaussian case, new methods of dealing with sequential data are required with non-standard models. Recently, there has been renewed interest in simulation-based techniques. The basic idea behind these techniques is that the current state of knowledge is encapsulated in a representative sample from the appropriate posterior distribution. As time goes on, the sample evolves and adapts recursively in accordance with newly acquired data. We give a critical review of recent developments, by reference to oil well monitoring, ion channel monitoring and tracking problems, and propose some alternative algorithms that avoid the weaknesses of the current methods.

Keywords: Markov Chain Monte Carlo, Importance resampling, Recursive filters, Condensation algorithm, SIR filter, MHIR filter.

1 Introduction

The use of Markov Chain Monte Carlo (MCMC) methods in Statistics, over the last decade or so, has produced profound changes in the ways in which data are modelled and interpreted (Gilks *et al.*, 1993, 1996; Smith and Roberts, 1993; Besag and Green, 1993). Although some caution may still be appropriate, since it seems that Monte Carlo methods are particularly vulnerable to undetected programming errors, there is now a substantial body of experience with computer implementation of these methods. By applying good programming practice and by insisting on scrupulous validation against known distributional results, MCMC techniques can now be applied with confidence to a wide range of important new statistical applications.

The situation with regard to dynamic statistical models and evolving data sets is considerably less happy. Dynamic models arise naturally in many scientific, industrial and financial problems — indeed in any situation where we wish to analyse the state of a system as it evolves in time, but from which only partial information is available. The sequence of past states in a dynamic model forms a parameter set to which new parameters are continually being added. A direct application of MCMC techniques to models in which the dimension of the parameter space grows with time rapidly runs into difficulty. As time progresses the historical parameter space expands and becomes increasingly difficult to explore with a Markov chain. Thus any attempt at on-line implementation will eventually be defeated.

Inference problems for dynamic models fall into three broad categories: smoothing, filtering and prediction. In a smoothing problem, the aim is to reconstruct past states of the system by using all the data which have been accumulated so far. Filtering is concerned with updating the present state of knowledge, and prediction with drawing inferences about the state of the system in the future. In many applications, filtering is the key problem, since it forms the basis of prediction and control. We therefore focus on it here.

2 The Basics of Bayesian Filtering

A common assumption in many dynamic models is that the state space evolves as a Markov process. Successive application of Bayes' theorem then indicates that inferences about the current state, given the most up-to-date information available from observations, can be updated in a sequential manner as new information becomes available.

Specifically, suppose the state of the dynamic system at time t, can be represented by the state vector $\theta_t \in \mathbb{R}^p$, and that the state at time $t+1$ is related to the previous state by the state model

$$\theta_{t+1} = f_t(\theta_t, w_t). \tag{1}$$

Here $f_t : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p$ is the known system transition function and $w_t \in \mathbb{R}^q$ is a noise term with known distribution.

At each discrete time point, an observation $x_t \in \mathbb{R}^r$ is obtained, related to the state vector by

$$x_t = h_t(\theta_t, v_t),$$

where $h_t : \mathbb{R}^p \times \mathbb{R}^s \to \mathbb{R}^r$ is the known measurement function and $v_t \in \mathbb{R}^s$ is another noise term whose known distribution is independent of both the system noise and time. We write $\mathbf{x}_t$ for $(x_1, ..., x_t)$, the available information up to time t, and assume the PDF of θ_0, the initial state of the system, is known so that $p(\theta_0|\mathbf{x}_0) = p(\theta_0)$. We then wish to obtain the PDFs $p(\theta_t|\mathbf{x}_t) : t \geq 1$, which are given by the update equations:

$$p(\theta_{t+1}|\mathbf{x}_t) = \int p(\theta_{t+1}|\theta_t)p(\theta_t|\mathbf{x}_t)\, d\mu(\theta_t) \tag{2}$$

and

$$p(\theta_{t+1}|\mathbf{x}_{t+1}) = \frac{p(x_{t+1}|\theta_{t+1})p(\theta_{t+1}|\mathbf{x}_t)}{p(x_{t+1}|\mathbf{x}_t)}, \tag{3}$$

where

$$p(x_{t+1}|\mathbf{x}_t) = \int p(x_{t+1}|\theta_{t+1})p(\theta_{t+1}|\mathbf{x}_t)\, d\mu(\theta_{t+1}). \tag{4}$$

These equations therefore represent the formal solution to the sequential estimation problem. Methods for solving them form the subject of the rest of the paper.

If the functions f_t, h_t, above are linear and the noise terms w_t, v_t, are Gaussian, then the solution to (2)–(4) is the Kalman filter (Jazwinski, 1973). This represents the uncertainty about the state of the system by the means and covariances of a multivariate Gaussian distribution.

However, in many practical applications, these assumptions are implausible. In particular, the observation process will often be non-linear. An alternative approach in such cases is the extended Kalman filter (EKF) (Jazwinski, 1973), in which the updated measurements are linearised about the predicted state, permitting the Kalman filter to be applied approximately. This algorithm and its refinements have proved popular, particularly in the field of object tracking. However, the Gaussian approximation to the density of the underlying state, inherent in the EKF, will often prove to be inadequate, causing the update procedure to become unstable (Johnson *et al.*, 1983; Weiss and Moore, 1980; Aidala, 1979).

In addition, the linearisation presents problems. Such problems tend to manifest themselves in the early stages of tracking, but they have the potential to occur at any stage. Errors due to linearisation tend to be self-perpetuating problem and can preclude an accurate estimate ever being found. To avoid this problem, the filter needs to be initialised with informative prior means and covariances, yet in practice such information is frequently unavailable. Early attempts to tackle the initialisation problem within the context of object tracking (Aidala and Hammel, 1983; Aidala, 1979; Lindgren and Gong, 1978) have not proved entirely successful (Nardone *et al.*, 1984; Aidala and Nardone, 1982). A recent development (Moon and Stevens, 1996) is more promising, but all such approaches are highly problem specific.

Other approximate analytical methods include the Gaussian sum filter (Alspach and Sorenson, 1972), methods that involve approximating the first two moments of the density (Masreliez, 1975; West *et al.*, 1985) and numerical approaches which evaluate the required probability density function over a grid in the state space (Bucy, 1969; Kramer and Sorenson, 1988; Sorenson, 1988; Kitagawa, 1987; Pole and West, 1990). However, none of these techniques represents a universal algorithm, since they each have to be extensively modified to tackle the problem in hand. For example, methods that evaluate the probability density over a grid in the state space first require the grid to be specified, which is a non-trivial problem in a multi-dimensional space. To avoid misleading results, a large number of grid points will in general be necessary. In addition, a non-trivial computation must be performed at each point.

In this paper we will focus on an alternative class of filters in which theoretical distributions on the state space are approximated by simulated random measures. The first goal in filter design is to produce a compact description of the posterior distribution of the state given all the observations available so far. A basic requirement is that this description should be readily updated as new data become available. A mechanism has therefore to be devised which enables the approximating random measure to evolve and adapt.

3 Simulation based filters

Simulation based filters have a long history in the engineering literature, dating back to the work of Handschin and Mayne (1969); Handschin (1970); Akashi and Kumamoto (1977). Doucet (1998) provides a comprehensive review of the material. Since the Kalman filter is essentially a Bayesian update formula, the theory of Bayesian time series analysis is directly relevant (West and Harrison, 1997).

We take as our starting point the filter developed by Gordon (1993); Gordon *et al.* (1993). The essence of the method is contained in a paper by Rubin (1988) who proposed the Sampling Importance Resampling (SIR) algorithm for obtaining samples from a complex posterior distribution without recourse to MCMC. In the simple non-dynamic case described by Rubin (1988), the method consists of sampling n observations from the prior distribution, attaching weights to the sampled points according to their likelihood, and then sampling with replacement from this weighted discrete distribution. As $n \to \infty$, the resulting set of values then approximates a sample from the required posterior (Smith and Gelfand, 1992). In the dynamic version, proposed by Gordon *et al.* (1993), the SIR algorithm is applied repeatedly as new data are acquired. One can think of the sample points in a SIR filter as a set of particles which move according to the state model and multiply or die depending on their 'fitness' as determined by the likelihood function. The SIR filter is one of a large class of particle filters. Filters of this type have been developed independently by other research groups (Kitagawa, 1996; Isard and

Blake, 1996).

3.1 SIR Filter

The SIR filter proceeds as follows. Assume we have a sample $(\theta_t^i)_{i=1,\ldots,n}$ from $p(\theta_t|\mathbf{x}_t)$.

(a) **Prediction:** Simulate $\tilde{\theta}_{t+1}^i = f_t(\theta_t^i, w_t^i)$, independently for each $i = 1, \ldots, n$, using the state model (1).

(b) **Likelihood:** Upon receipt of observation x_{t+1}, for each value $\tilde{\theta}_{t+1}^i$ calculate the corresponding likelihood $p(x_{t+1}|\theta_{t+1}^i)$. Denote the set of likelihood values, normalised to sum to 1, by $(q_{t+1}^i)_{i=1,\ldots,n}$.

(c) **Update:** Draw a random sample of size n from the discrete distribution taking values $(\tilde{\theta}_{t+1}^i)_{i=1,\ldots,n}$ with probabilities $(q_{t+1}^i)_{i=1,\ldots,n}$. This is an approximation to a sample from $p(\theta_{t+1}|\mathbf{x}_{t+1})$.

We now outline some of the problems that may occur. The first problem we consider is *sample impoverishment*. To illustrate this we consider a degenerate case in which the state is fixed.

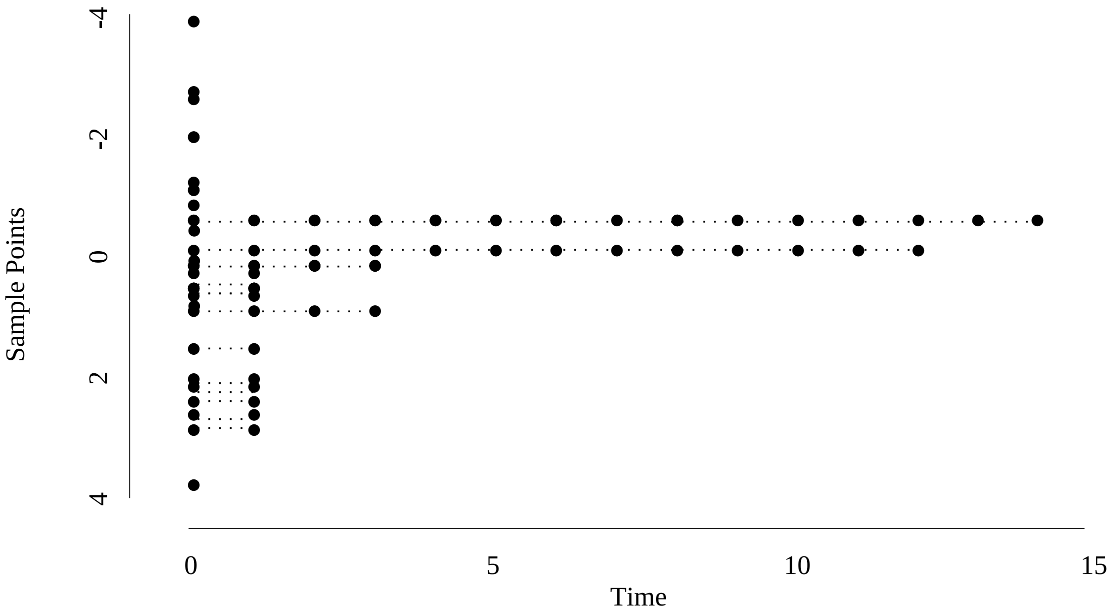

Figure 3.1: Illustration of sample depletion in the SIR filter. Leftmost points are a sample of size 30 from $N(0, 4)$, the prior distribution. Dotted lines join those support points which survive each update.

EXAMPLE 3.1 *Sequential estimation of normal mean* The objective is to estimate the posterior mean of a univariate normal distribution when the observations arise sequentially.

In the notation of Section 2, the state model is $\theta_{t+1} = f_t(\theta_t, w_t) = \theta_t$, in other words the unknown state does not change with time. The observations are given by $x_t = h_t(\theta_t, v_t) = \theta_t + v_t$, for $t = 1, \ldots, 15$, where $v_t \sim N(0, 1)$. The prior density of θ_0 is taken to be $N(0, 4)$. For illustration we fix $\theta_0 = 1$. The SIR algorithm is initialised with a sample of size 30 from $p(\theta_0)$.

Figure 3.1 shows what happens. The upper tail of the sample values is immediately lost because the initial observation happens to be negative. The remaining support points are gradually lost with repeated sampling. After thirteen updates the support has degenerated to a single point. The estimate of the mean based on this surviving point is 0.51, with an estimated standard error of zero. This compares with the actual posterior distribution at this stage, which is $N(1.03, 0.075)$. Although such extreme behavior is unlikely for genuinely dynamic state models, the SIR filter typically suffers from underdispersion to a greater or lesser extent.

The second problem with the SIR filter occurs when outlying observations are encountered. The effect of an outlying observation is to produce a likelihood which is centered in the tail of the prior distribution. Since this tail is represented only sparsely by sample points in the SIR filter, an exceptionally large sample from the prior will be needed to yield a good support for the posterior distribution.

These problems are clearly related: a poor representation of the prior distribution may cause an otherwise plausible data point to appear an outlier; conversely, a real outlier may cause severe depletion of the sample support. Nevertheless, we separate them, as specific proposals have been targeted at each.

3.2 *Jittering, prior boosting and editing*

In order to alleviate problem of sample impoverishment, Gordon *et al.* (1993) give an *ad hoc* rule for adding a small amount of Gaussian noise, or jitter, to each sample point at each time step. If one point is replicated in the posterior r times, it is now replaced by r closely adjacent points.

Jittering therefore smoothes out the posterior density, using a Gaussian kernel. Choosing the jitter variance is thus equivalent to choosing the smoothing parameter in density estimation, and there is a corresponding variance/bias tradeoff to be made. Standard rules of thumb can be used to choose the degree of smoothing (e.g. Silverman, 1986).

A related method, where a variance correction term in the jittering is introduced, is suggested by Acklam (1996). This aims to partially compensate for the extra variance induced by the jitter, by shrinking particles towards their sample mean. Acklam also proposes the equivalent of local density estimation, whereby the jitter variance varies from particle to particle.

Besides expanding the sample support, jittering also reduces the risk of filter divergence, since, if the sample of states has a larger dispersion, a new outlying data point will appear less extreme. In this respect, a link can be made to methods which have been proposed for avoiding EKF divergence by artificially increasing the variance.

Another approach to sample depletion was originally proposed by Rubin (1987). At the prediction stage of the SIR filter, instead of generating the usual n points, we generate κn points. The likelihood of each of these is calculated, and then n are resampled in the update step in the usual way. Typically $\kappa = 10$.

Clearly, the idea behind this is that by boosting the number of samples available from the prior, the number of distinct points in the posterior will be increased. While this is generally true, a straightforward comparison with the SIR filter with κn sample points shows that prior boosting increases the variance of the Monte-Carlo estimate of any quantity of interest. Since the computational burden is comparable, we therefore prefer the ordinary SIR filter.

Prior editing was proposed by Gordon *et al.* (1993) as a way of addressing the outlier problem. Starting from a sample $(\theta_t^i)_{i=1,\ldots,n}$, from $p(\theta_t|\mathbf{x}_t)$, the idea is to repeatedly sample from $p(\theta_{t+1}|\theta_t^i)$ for $i = 1, \ldots, n$ and only accept the simulated value if its likelihood $p(x_{t+1}|\theta_{t+1})$ is high.

Clearly, this procedure relies on being able to delay the state estimate for one time step, which may prove problematic in some applications. The risk of filter divergence is reduced, by focusing the sample $(\theta^i_{t+1})_{i=1,\ldots,n}$ in a region of relatively high probability in relation to x_{t+1}. A disadvantage of the procedure is that $(\theta^i_{t+1})_{i=1,\ldots,n}$ will be a very approximate sample from $p(\theta_{t+1}|\mathbf{x}_{t+1})$. If ongoing estimates of $p(\theta_{t+1}|\mathbf{x}_{t+1})$ are required, they will have to be taken from samples obtained without editing.

3.3 *Auxiliary variables*

Based on a sample $(\theta^i_t)_{i=1,\ldots,n}$ from $p(\theta_t|\mathbf{x}_t)$ the density $p(\theta_{t+1}|\mathbf{x}_t)$ in (2) can be estimated by

$$n^{-1}\sum_{i=1}^{n} p(\theta_{t+1}|\theta^i_t).$$

Substituting this estimate into (3) gives an estimate for $p(\theta_{t+1}|\mathbf{x}_{t+1})$ of the mixture form

$$K\sum_{i=1}^{n} p(x_{t+1}|\theta_{t+1})p(\theta_{t+1}|\theta^i_t),$$

where K is a normalising constant. A standard method for sampling from a mixture distribution is to introduce an auxiliary variable which acts as an index of the mixture component (Besag and Green, 1993). The base measure for the joint density of the variable θ_{t+1} and the index z is the product $\mu \times \nu$ where ν is standard counting measure. With respect to this measure the joint density of $(\theta_{t+1}, z|\mathbf{x}_{t+1})$ is proportional $p(x_{t+1}|\theta_{t+1})p(\theta_{t+1}|\theta^z_t)$. Once a density is known up to normalising constant, MCMC methods can be employed to obtain approximate samples. Berzuini *et al.* (1996) have adopted this approach using the Metropolis-Hastings 'independence' sampler. They take as a proposal distribution a randomly selected index and, conditional on the index $z = i$, a sample from $\theta_{t+1}|\theta^i_t$. Our experience with this method suggests that the sampler may fail to mix effectively.

Another method of this type has been suggested by Pitt and Shephard (1997). They use Sampling Importance Resampling to sample from the pair (θ_{t+1}, z) conditional on $\mathbf{x}_{t+1}$. By sampling the index independently from a distribution which is closely adapted to the weights of the mixture components and by using good approximations to the densities within each mixture component, they are able to demonstrate a performance superior to that of the standard SIR filter. However, it is important to note that the SIR filter samples the index *systematically*, so that each mixture component is represented exactly once. This is a positive feature of the SIR filter. As we shall see later, systematic sampling can play an important role in variance reduction.

Both of these methods can be extended by replacing θ_{t+1} by $\{\theta_{t+1}, \theta_{t+2}, \ldots, \theta_{t+\tau}\}$. The filter then aims to sample from the posterior distribution of these variables at time $t+\tau$ given $\mathbf{x}_{t+\tau}$. The potential benefit of this approach is that a poor approximation of $p(\theta_t|\mathbf{x}_t)$ will have less of an effect on the estimate of $p(\theta_{t+\tau}|\mathbf{x}_{t+\tau})$ as τ increases.

4 Random measures

Particle filters work by providing a discrete approximation to the PDF which can be easily updated to incorporate new information as it arrives. Our interest will be in approximations which consist of a set of random locations in the state space $(s^i)_{i=1,\dots,n}$, termed the *support*, and a set of associated weights $(m^i)_{i=1,\dots,n}$ summing to 1. The support and the weights together form a *random measure*.

The objective is to choose measures so that

$$\sum_{i=1}^{n} g(s^i)m^i \approx \int g(\theta)p(\theta)\,d\mu(\theta) \tag{5}$$

for typical functions g of the state space, in the sense that the left-hand side of (5) converges (in probability) to the right-hand side as $n \to \infty$.

The simplest example of a random measure is obtained by sampling $(s^i)_{i=1,\dots,n}$ independently from $p(\theta)$, and giving equal weights $m^i = n^{-1}; i = 1, \dots, n$. The left-hand side of (5) then becomes the sample average $\sum_{i=1}^{n} g(s^i)/n$. Importance sampling (Hammersley and Handscomb, 1964) provides a more general example by sampling $(s^i)_{i=1,\dots,n}$ from another PDF $f(x)$ and attaching importance weights $m^i = Ap(s^i)/f(s^i)$, where $A^{-1} = \sum_{i=1}^{n} p(s^i)/f(s^i)$. We develop these ideas further for particle filters in Section 6. First, however, we describe the basic SIR filter as a random measure, as a prelude to evaluating various modifications.

Start by simulating a sample $(s_0^i)_{i=1\dots n}$ from $p(\theta_0)$. In other words, start from a random measure with equal weight on each of the n sample values. Assume that at stage t we have a random measure $(s_t^i, m_t^i)_{i=1,\dots,n}$, approximating $p(\theta_t|\mathbf{x}_t)$.

(a) **Prediction:** Estimate the density $p(\theta_{t+1}|\mathbf{x}_{t+1})$, up to a normalising constant K, by the n-component mixture

$$K\sum_{i=1}^{n} m_t^i p(\theta_{t+1}|s_t^i)p(x_{t+1}|\theta_{t+1}). \tag{6}$$

Take a systematic sample from this density, with exactly one sample point from each of the n components. In the ith component, sample a support point $\tilde{s}_{t+1}^i = f_t(s_t^i, w_t^i)$ from the system model.

(b) **Likelihood:** Calculate importance weights

$$\tilde{m}_{t+1}^i = \frac{m_t^i p(x_{t+1}|\tilde{s}_{t+1}^i)}{\sum_{i=1}^{n} m_t^i p(x_{t+1}|\tilde{s}_{t+1}^j)}. \tag{7}$$

(c) **Update:** Resample from the random measure $(\tilde{s}_{t+1}, \tilde{m}_{t+1})_{i=1,\dots,n}$ to obtain an equally weighted random measure. In other words, sample n times, independently with replacement, from the set $(\tilde{s}_{t+1}^i)_{i=1,\dots,n}$, with associated probabilities $(\tilde{m}_{t+1}^i)_{i=1,\dots,n}$, to obtain the random measure $(s_{t+1}^i, m_{t+1}^i)_{i=1,\dots,n}$, with $m_{t+1}^i = n^{-1}, i = 1, \dots, n$.

Before attempting to improve the algorithm, it is worth emphasing that our fundamental objective is to produce accurate Monte Carlo approximations to the succession of integrals which arise in Bayesian calculations. For accurate Monte Carlo integration, it is essential to eliminate unnecessary randomness and to make careful choices for proposals in importance sampling. The randomness introduced by sampling with replacement, at stage (c) above, can be reduced by a systematic algorithm. The purpose of resampling is to convert a random measure with non-uniform weights to one where the weights are equal. To take a slightly more general case, suppose that $(m^i)_{i=1,\dots,\ell}$ is a set of weights summing to 1 and $(s^i)_{i=1,\dots,\ell}$ is an associated set of support points. The aim is to sample the ith support point N_i times so that the expected value of N_i is nm_i. Sampling with replacement achieves this, but the variables $(N_i)_{i=1,\dots,n}$ are multinomially distributed. With the following algorithm (Carpenter *et al.*, 1998) the variables never differ from their required expected value by more than 1. In the resampling application above, ℓ equals n.

ALGORITHM 1 Systematic sampling

$T = unif(0, n^{-1}); j = 1; Q = 0; i = 0$
do while $T < 1$
  if $Q > T$ *then*
    $T = T + 1/n$; *output* s^i
  else
    pick k *in* $\{j, \dots, \ell\}$
    $i = s^k$
    $Q = Q + m^i$
    switch (s^k, m^k) *with* (s^j, m^j)
    $j = j + 1$
  end if
end do

The algorithm treats the weights as contiguous intervals of $(0, 1)$. These intervals are randomly ordered, and the number of grid points $\{T + k/n\}$ in each interval is then counted. Other systematic algorithms of this type have been proposed by Crisan and Lyons (1997) and Liu and Chen (1997).

If our only objective is to minimise sample impoverishment, this is most easily achieved by eliminating stage (c), that is, simply to take $s_{t+1} = \tilde{s}_{t+1}$ and $m_{t+1} = \tilde{m}_{t+1}$ and carry the modified weights forward, without resampling. It is easy to show that an estimator based on the resampled measure is less efficient than one based on $(\tilde{s}^i_{t+1}, \tilde{m}^i_{t+1})_{i=1,\dots,n}$. This may suggest that it is always a good idea to carry the weights forward. However, resampling presents an opportunity to remove particles with very small weights and to concentrate particles in regions of the state space which are well supported by the data. In other words, the benefits of resampling may emerge at a later stage.

This raises the question, when should we resample, and when should we carry forward the weights? The question has been addressed by Liu and Chen (1997) and Liu *et al.* (1996) who propose an *ad hoc* rule based on the variance of the weights $(\tilde{m}^i_t)_{i=1,\dots,n}$. To be specific, the question is whether $T_1 = \sum_{i=1}^n \tilde{m}^i_t y^{i,1}$ or $T_2 = n^{-1} \sum_{i=1}^n \sum_{j=1}^{N_i} y^{i,j}$ has the smaller variance, where $\mathbf{E}\, N_i = n\tilde{m}^i_t, i = 1, \dots, n$, and $(y^{i,j})$ are independent random variables involving the future state of the system. If $\mathbf{E}\, y^{i,j} = \mu_i$, and assuming for simplicity that $\operatorname{Var} y^{i,j} = \sigma^2$, then resampling is beneficial, that is, $\operatorname{Var} T_2 \leqslant \operatorname{Var} T_1$ when

$$\frac{1}{n^2} \operatorname{Var} \left(\sum_{i=1}^n \mu_i N_i \right) \leqslant \sigma^2 \sum_{i=1}^n (\tilde{m}^i_t - \bar{m}_t)^2,$$

where $\bar{m}_t$ is the average weight. It follows that the decision about whether to resample depends on the method of resampling. If the numbers $(N_i)_{i=1,\dots,n}$ have small variability, then resampling may be worthwhile. However, the choice also depends on the variability of the values $(\mu_i)_{i=1,\dots,n}$ and the variance σ^2, quantites which involve the future state of the system. In general, if the weights are roughly even, and the system noise is small compared to the variance of the posterior at the previous time step, then it is better not to resample. In particular, if there is no system noise, so that $\sigma^2 = 0$, resampling is always inefficient.

We should note that there is intermediate ground between resampling and carrying forward the weights. Resampling can be carried out using *tempered* weights: for example, using weights proportional to the square root of the original, i.e. $w_t^i \propto \sqrt{m_t^i}$. The resampled points are then carried forward with weights proportional to m_t^i/w_t^i. Similar techniques have been proposed in MCMC sampling to avoid problems in sampling from highly peaked densities.

Efficiency gains are also possible at the prediction stage (b). Using estimates of the posterior weights of the mixture components, Algorithm 1 can be used to determine how many representatives should be in each component. Samples are then drawn within the mixture components according to a proposal distribution and the resulting particles are then reweighted appropriately. See Carpenter *et al.* (1998).

4.1 Better Proposals

Although it is unrealistic to use MCMC to sample the posterior distribution of the complete state history, under certain circumstances MCMC moves can be introduced in particle filtering. These moves may be successful in preventing sample impoverishment. Before discussing this possibility we note the following. For $i = 1, \ldots, n$, suppose that α_i is sampled from a density $p(\alpha)$ and β_i is sampled from $\pi(\beta|\alpha_i)$, the transition density of a Markov chain with equilibrium density π. Define $m_i = \pi(\alpha_i)/p(\alpha_i)$, then the random measure $(m^i, \beta_i)_{i=1,\ldots,n}$ approximates the distribution with density π.

A variant of the same argument shows that if $(\tilde{\theta}^i, n^{-1})_{i=1,\ldots,n}$ is an approximating uniform random measure for the posterior of the state history $\theta = (\theta_0, \ldots, \theta_t)$ at time t then so is the uniform measure with support points sampled from $\pi(\theta_t|\tilde{\theta}_t^i)$, where the transition density $\pi(\cdot|\cdot)$ preserves the posterior distribution.

Clearly, this approach will combat the depletion of support points in the approximating measure. Moreover, if the transition density is a continuous density, all the points in the sample will now be different. As usual in MCMC, the efficiency depends on the appropriateness of the transition density. A particularly attractive approach, where it is possible, is to use a Gibbs transition kernel (MacEachern *et al.* (1998)).

To accommodate arbitrary transitions it is necessary to store the whole history of the process up to time t. As we shall see in the next example, this can be avoided if the transition kernel only depends on a fixed set of summary statistics, or only upon the last τ time points.

EXAMPLE 4.1 *Bearings only tracking.* A classic example of non-linear filtering is bearings only tracking. Figure 1 shows a typical trajectory generated from the following linear system,

$$x_t = \begin{pmatrix} \nu_t \\ \dot{\nu}_t \\ \xi_t \\ \dot{\xi}_t \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \nu_{t-1} \\ \dot{\nu}_{t-1} \\ \xi_{t-1} \\ \dot{\xi}_{t-1} \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} w_{t-1}^{(1)} \\ w_{t-1}^{(2)} \end{pmatrix}, \tag{8}$$

$t = 1, \ldots, 24$, where $\theta_t = (\nu_t, \xi_t, \dot{\nu}_t, \dot{\xi}_t)$ is the position and velocity at time t and $w_t^{(1)}, w_t^{(2)}$ are independent normally distributed random variables with means 0 and variances σ_w^2 distribution. At each time point, t, we observe a noisy bearing or angle, x_t, given by

$$x_t = \tau_t + v_t,$$

where $v_t \sim N(0, \sigma_v^2)$ and $\tau_t = \tan^{-1}(\xi_t/\nu_t)$.

We seek to reconstruct the trajectory given the system model, observed bearings and a prior distribution on the initial position and velocity. In particular, suppose we observe t bearings.

Then we are interested in estimating the posterior density of the trajectory, or in other words the joint posterior distribution of

$$\theta = (\theta_0, \theta_1, \ldots, \theta_t)\,, \quad \text{given} \quad \mathbf{x} = (x_1, x_2, \ldots, x_t)\,.$$

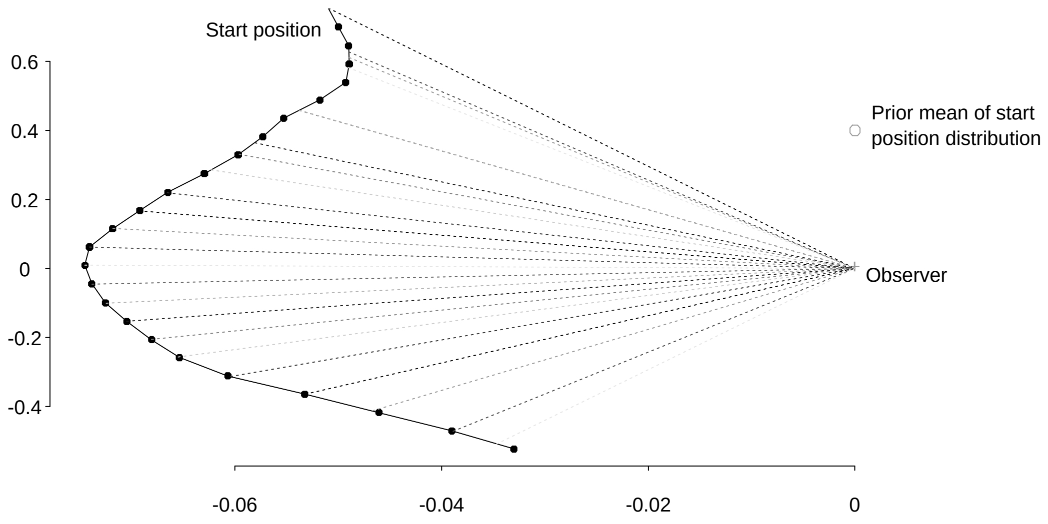

Figure 4.2: Typical trajectory simulated from (8). Dotted lines show observed bearings

The posterior density of the track $(\theta_0, \theta_1, \ldots, \theta_t)$ is proportional to

$$p(\nu_0, \xi_0, \nu_1, \xi_1)\left(\prod_{k=1}^{t} p(z_k|\tau_k)\right)\exp(-A_t(\theta)),$$

where

$$A_t(\theta) = \frac{1}{2\sigma_w^2}\sum_{k=2}^{t}[(\nu_k - 2\nu_{k-1} + \nu_{k-2})^2 + (\xi_k - 2\xi_{k-1} + \xi_{k-2})^2].$$

Note that scaling the track by multiplying each θ_t by a constant λ does not affect the likelihood since none of the angles change. It multiplies $A_t(\theta)$ by λ^2 and affects the term $p(\nu_0, \xi_0, \nu_1, \xi_1)$. These factors can be incorporated into the filter by extending the *signature* of each particle. We do this by storing for each particle an extended vector $(\nu_0, \xi_0, \nu_1, \xi_1, A_t(\theta), \nu_t, \xi_t)$. The MCMC scale move, when it is made, is a Gibbs move, sampling from a modified Gamma distribution. For the implementation details and an analysis of the effectiveness of the move, see Fearnhead (1998).

5 Assessing sample impoverishment

The foregoing discussion has described several proposals for improving the reliability of particle filters by reducing the rate of sample degeneracy. To compare the proposals, we need a measure of the effective sample size. This is the sample size that would be required for a simple random sample from the target posterior density to achieve the same estimating precision as the random measure provided by the particle filter. Since some properties of the state distribution may be estimated well, and some poorly, the effective sample size will depend on the quantity being estimated.

5.1 *The effective sample size in importance sampling*

We start by considering the loss of efficiency resulting from the use of importance sampling at each stage in the SIR filter. In particular we will look at an approximation to the effective sample size suggested by Liu (1996) and Neal (1998), which we show can be arbitrarily wrong. Suppose we are interested in estimating μ, the expected value of $g(\theta)$, where the random variable θ has density function $\pi(\theta)$ and g is some function of interest. Using the average of $g(\theta)$ for a random sample of size m from the distribution π, our estimate will have variance $\sigma_\pi^2 = \text{Var}_\pi g(\theta)/m$. Now suppose we wish to estimate μ using a sample of size n from a proposal density $p(\theta)$. Denote the variance of this new estimate by σ_p^2. Then the effective sample size is the value of m for which $\sigma_\pi^2 = \sigma_p^2$. Liu (1996) has suggested that σ_p^2 can be approximated by $\text{Var}_\pi g(\theta)(1 + \text{Var}_p r(\theta))/n$, where $r = \pi/p$. The effective sample size is then $n/(1 + \text{Var}_p r(\theta))$. However, this result should be used with caution. Even in simple cases, where p is a Gaussian density, the formula can be misleading; see the Appendix, where a more robust estimate is given.

If our objective is to provide a discrete approximation to a given distribution, using a limited amount of computer storage, importance sampling will generally be inefficient. Regularised sampling by various means can produce substantial economies. For example, suppose we want to represent the uniform distribution on $(0, 1)$. We could do this using a simple random sample of size n, in which case the the sample average would have variance $1/(12n)$. Alternatively, we could sample n points in a stratified manner, selecting exactly one point uniformly in each interval $(k/n, (k+1)/n)$ for $k = 0, \ldots, n-1$. The variance of the average of these points would then be $1/(12n^3)$. The effective sample size of this 'random measure' would then be n^3. Although this effect decreases with increasing dimension (Fishman, 1996), there is evidence that regularised sampling can be effective in high dimensions when the integral is dominated by variation in a few components (Caflisch *et al.*, 1997). In the bearings only example, we conjecture that the scale factor plays that role.

5.2 *Monte-Carlo estimate of effective sample size*

In practice efficiency at a single update stage is of less interest than the efficiency of the filter as a whole, after a number of time steps have elapsed. Carpenter *et al.* (1998) propose a Monte-Carlo estimate of effective sample size, which is basically an application of the classical 'analysis of variance'. Suppose that we are interested in estimating μ_t the posterior mean of $g(\theta_t)$ given $\mathbf{x}_t$. In a particle filter, the posterior distribution is represented by the random measure $(\tilde{s}_t^i, m_t^i)_{i=1,\ldots,n}$, prior to resampling. Let

$$z_t = \sum_{i=1}^{n} m_t^i g(\tilde{s}_t^i), \qquad v_t = \sum_{i=1}^{n} m_t^i g^2(\tilde{s}_t^i) - z_t^2$$

be the filter estimates of μ_t and σ_t^2, the posterior variance of $g(\theta_t)$. Note that, if μ_t is estimated by using the average value of $g(\theta_t)$ in a simple random sample of size m from $\pi(\theta_t|\mathbf{x}_t)$, the estimate will have variance σ_t^2/m.

The proposal is then as follows:

1. Run the filter independently K times, obtaining K independent replicates, each based on n particles.

2. For each replicate, at step t, calculate z_t^j and $v_t^j, j = 1 \ldots K$.

3. Calculate $\bar{z}_t$ and $\bar{v}_t$, the average values over the K replicates.

4. The *effective sample size* is then $K\bar{v}_t / \sum_{j=1}^{K}(z_t^j - \bar{z}_t)^2$.

To see this, we equate two estimates of the variance of z_k: one based on the variance between replicates and the other based on the notional variance that an estimate would have if it was a sample average of a simple random sample of size m, *i.e.*

$$K^{-1}\sum_{j=1}^{K}(z_t^j - \bar{z}_t)^2 = \frac{\bar{v}_t}{m}.$$

The effective sample size is then obtained by solving for m.

We advocate the use of this diagnostic generally, in assessing the performance of Monte Carlo filters. The smaller the effective sample size is, the less reliable the filter is. In principle, a Bayesian filter should be assessed by looking at its performance averaged over the population of trajectories generated by the system model. However, for non-linear problems it may happen that most of the trajectories are simple to filter and only a few are 'difficult cases'. It is therefore helpful to see how the filter performs for typical examples of these difficult cases.

The integrated correlation time in Markov chain Monte Carlo (MCMC) calculations in non-dynamic problems (Gilks *et al.*, 1996) and the effective sample size play similar roles. Neither of these diagnostics is designed to check for convergence to the *right* distribution. A noisy biased filter may have a large effective sample size but the sample may not have come from the correct distribution. To check for bias, the proposed particle filter will need to be compared with filters which are known to perform correctly.

6 Parametric particle filters

Statistical models in which the behaviour of a process variable is controlled by a hidden Markov chain are an important class of problems to which particle filters can be applied. In many cases, it is reasonable to assume that Bayesian updating can be carried out explicitly once the state of the hidden chain is known. For such cases, a special class of filters can be introduced, namely particle filters which are tagged by parametric information.

METHOD 1 In the linear Gaussian case, when the state of the chain is known the Kalman filter can be applied to obtain explicit update formulae in terms of means and variances. For the parametric particle filter, the posterior distribution at time t is represented by a random measure $(s_t^i, m_t^i)_{i=1,\ldots,n}$, where now the 'signature' of s_t^i incorporates the state γ_t of the hidden Markov chain and α_t, the associated posterior mean and variance of the process variable at time t. The signature may also contain information about past states of the system if required.

Since the chain has K states an obvious strategy is to split each particle into K new ones, with support points $(\alpha_{t+1}^{i,j}, j)_{j=1,\ldots,K}$ where $\alpha_{t+1}^{i,j}$ is the posterior mean and variance of the process variable at time $t+1$. These parameters are calculated by conditioning on the new observation, x_{t+1}, the previous parameters, α_t^i, and the updated hidden state $\gamma_{t+1} = j$. The updated weights are then proportional to

$$m_{t+1}^{i,j} = m_t^i p(x_{t+1}|\gamma_{t+1} = j, \alpha_t^i)\mathbf{P}\left(\gamma_{t+1} = j|\gamma_t^i\right) \tag{9}$$

for $j = 1, \ldots, K$. Clearly, to retain computational tractability, we must devise a method of selecting n particles from this expanded set of nK particles. We term this the problem of *support economisation*.

To simplify notation we will drop subscripts and consider the problem of constructing a random measure $(\tilde{s}^i, \tilde{m}^i)_{i=1,\ldots,L}$ which approximates $(s^i, m^i)_{i=1,\ldots,L}$ such that (i) $\mathbf{E}\,\tilde{m}^i = m^i$, (ii) not more than n of the $\tilde{m}^i$ are strictly positive, for some specified $n < L$n , and (iii) the measure is optimal in the sense of minimising the following discrepancy between the measures

$$\mathbf{E}\left(\sum_{i=1}^{L}(m^i - \tilde{m}^i)^2\right). \tag{10}$$

Fearnhead (1998) has shown that this can be achieved as follows. First solve for c in the equation $\sum_{i=1}^{L}\min(cm^i, 1) = n$. Support points for which $cm^i > 1$ are then retained with unchanged weights. If k points are retained then $n-k$ points are subsampled from the remaining $L-k$ points systematically using Algorithm 1, thus ensuring that n points in total are retained. Each of the latter class of retained points is given weight $1/c$.

METHOD 2 An alternative method is as follows. Assume that at time t we have a random measure $(s_t^i, m_t^i)_{i=1,\ldots,n}$ approximating $p(\theta_t|\mathbf{x}_t)$. As before, each support point s_t^i has two components: the state γ_t^i of the hidden Markov chain and α_t^i the parametric state of the process variable. The updated weights are given by (9). This time we consider an approximating set of weights $(m_t^i\beta_j, i = 1,\ldots,n, j = 1,\ldots,K)$, where $\sum_{j=1}^{K}\beta_j = 1$, and carry out the following steps:

1. Make K copies of each point s_t^i in the random measure at time t, obtaining $(s_t^{i,j})$ for $i = 1,\ldots,n; j = 1,\ldots,K$. Assign each point, $s_t^{i,j}$ a weight proportional to $m_t^i\beta_j$.
2. From this expanded random measure, sample n points, using Algorithm 1. Denote these by $\{\tilde{s}_t^{i,j}\}$. Note that some of the $s_t^{i,j}$ may appear in this new set more than once, and some not at all.
3. For each point, $\tilde{s}_t^{i,j}$, calculate the updated parameter values α_{t+1} and create a support point (α_{t+1}, j) with weight
$$\beta_j^{-1}\mathbf{P}\left(\gamma_{t+1} = j|\tilde{\gamma}_t^i\right)p(x_{t+1}|\alpha_{t+1}, \gamma_{t+1} = j).$$
4. Renumber the points to obtain the random measure $(s_{t+1}^i, m_{t+1}^i)_{i=1,\ldots,n}$ representing the posterior distribution at time $t+1$.

The second method has the advantage that it does not require calculating the posterior weighting factors for all of the nK potential new particles. If the mixture components are well chosen, it can be more efficient. However, where the true signal is constant for long periods, it tends to suffer from sample depletion. These points explain the superior performance of the second method in the target tracking in clutter example (Section 6.1), where the true signal follows a random walk. However, in the oil well and ion channel examples (Sections 6.2 and 6.3), the true signal is constant for long intervals, the first method is preferable.

We conclude this section by noting an interesting alternative approach. This is to represent the mean and covariance of the distribution at time t by a deterministically chosen set of points. These are then propagated in an analogous way to the SIR filter, and used to recover an estimate of the mean and covariance at time $t+1$. The details for the Gaussian case have been worked out by Julier *et al.* (1998).

6.1 Target tracking in clutter

The solid line in Figure 6.3 shows a signal of 1000 observations generated by a hidden Markov model. At each time point, 8 observations are received, one of which is the true state corrupted by Gaussian noise and the rest of which are Gaussian clutter. The hidden Markov model has

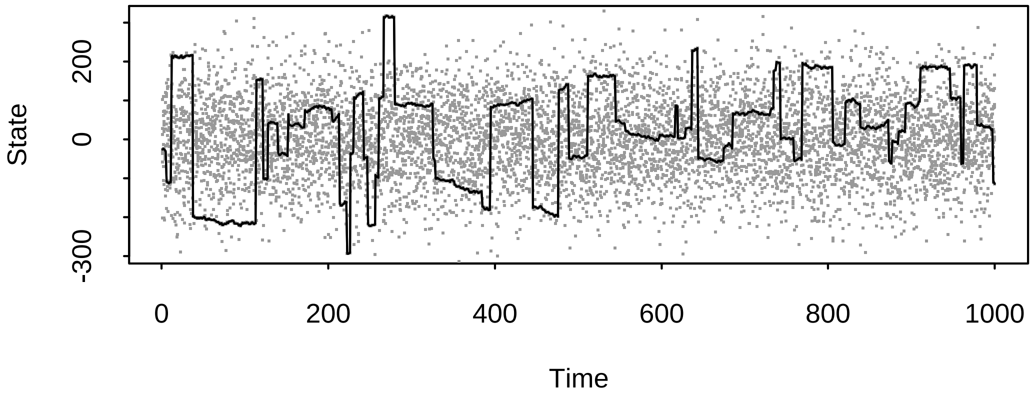

Figure 6.3: Target tracking in clutter: solid line is signal, generated according to (11); grey dots represent observations and clutter, generated according to (12).

2 state variables (β_t, γ_t). The signal at time t is represented by α_t. Together $\theta_t = (\alpha_t, \beta_t, \gamma_t)$ constitute the state of the system. The variables $\gamma_t, t = 1, \dots$ are independent and uniformly distributed on $(1, \dots, 8)$. They indicate which of the 8 observations is from the true state. The variables β_t are independent Bernoulli variables taking the value 1 with probability p_J. They determine the switching behaviour of the process α_t:

$$\alpha_{t+1} = \begin{cases} \alpha_t + w_t & \text{if } \beta_{t+1} = 0 \\ v_t & \text{if } \beta_{t+1} = 1 \end{cases} \tag{11}$$

where w_t and v_t are independent zero-mean Gaussian variables with variances σ_w^2 and σ_v^2 respectively. The observation vector $x_t = (x_t^{(1)}, \dots, x_t^{(8)})$ evolves as

$$x_t^{(j)} = \begin{cases} \alpha_t + h_t & \text{if } j = \gamma_t \\ g_t & \text{if } k \neq \gamma_t \end{cases} \tag{12}$$

where g_t and h_t are independent zero-mean Gaussian variables with variances σ_g^2 and σ_h^2 respectively.

Figure 6.3 was generated with the parameter values given in Table 1. Taking these parameter values as known, we attempted to recover the posterior distribution of the signal with the basic SIR filter, Methods 1 and 2, and two filters designed specifically for problems of this type: the IMM filter (Blom and Bar-Shalom, 1988) and the generalised pseudo-Bayes algorithm, with depth 2 (GPB(2)) (Tugnait, 1982).

Parameter	p_J	σ_w	σ_v	σ_h	σ_g
Value	$1/20$	$\sqrt{5}$	100	$\sqrt{0.05}$	100

Table 1: Parameter values used to generate Figure 6.3

The results are shown in Table 2 which gives the mean absolute deviation between the mean of the posterior distribution at time t and the true signal at time t.

For this example, the simple SIR filter always diverges before the end of this series, even with sample sizes as large as $20,000$. This is because there always comes a time when the signal jumps but the prediction stage of the filter has not generated any particles from the jump distribution with non-negligible likelihood. A simple fix when this happens is to hold $\beta = 1$ repeatedly till a particle with non-negligible likelihood is found. This is the modified SIR filter, which gives an idea of the performance of the basic SIR filter, were divergence not to occur.

Notice that the IMM filter recovers a very poor estimate of the signal, because it merges all the Gaussians making up the posterior at each time step, which is not appropriate. By contrast, the GPB(2) algorithm, which merges the Gaussian components of the posterior which came from the same state at time t, works well, and gives a gives a similar estimate to the stratified SIR filter with the same sample size. Both are slightly better than Method 1 with similar complexity. The advantage of Methods 1 and 2 is that they can be extended to non-linear, non-Gaussian systems, as in the example below, which the other methods do not handle. In addition, because Method 1 does not merge hypotheses, at any depth, and is constrained in the number of hypotheses retained only by the sample size, the 'memory' of the process, and hence its reliability, can be increased by simply increasing the sample size, whereas for the GPB algorithms, additional programming is necessary. The reason that Method 2 outperforms Method 1 in this example is discussed in section 6.

Filter	Complexity	Mean absolute deviation
Modified SIR	16	92.7
Modified SIR	100	64.5
IMM	16	86.4
GPB(2)	256	14.0
Method 1	256	18.3
Method 1	4096	14.2
Method 2	256	13.9

Table 2: Mean absolute deviation between true and recovered signal, for various filters described in the text. The complexity represents the number of Kalman filter updates (or equivalent) required by each filter at each time point, and is linearly related to speed.

6.2 Oil well log data

Figure 6.4 shows 4050 sequential measurements of nuclear magnetic response obtained while drilling for oil, kindly provided by Fitzgerald *et al* (Cambridge University Department of Engineering). The data contains information about the rock structure that is being drilled, and in particular about the boundaries between strata. It is important to be able to detect these boundaries accurately while drilling, in order to re-adjust the drilling pressure for the new rock type. Failure to do this may result in a 'blow-out'. Figure 6.4 suggests that the underlying signal could well be modelled as is a step function. Our analyses will seek to estimate these boundaries. Let α_t denote the signal at time t and let β_t be a hidden Bernoulli variable taking value 1 with probability p_J. This variable determines when a jump occurs. Our model is

$$\alpha_{t+1} = \begin{cases} \alpha_t & \text{if } \beta_{t+1} = 0 \\ w_t & \text{if } \beta_{t+1} = 1 \end{cases} \tag{13}$$

where w_t is taken to be Gaussian, mean μ_w and variance σ_w^2

This data set has been analysed by Ó Ruanaidh and Fitzgerald (1996), using an MCMC Gibbs sampler on the whole track. Before processing the data they remove the 40 greatest outliers. They assume that the noise process is independently sampled from a double exponential distribution.

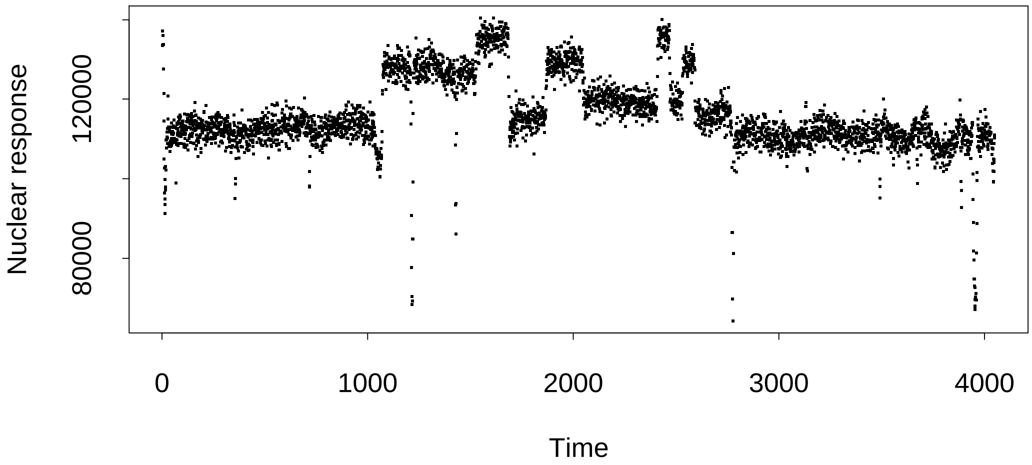

Figure 6.4: Plot of oil drilling data

We first obtained a crude estimate of the signal using a moving median with window length 63. Examining the residuals showed that, apart from the long lower tail, they were approximately normally distributed. They was also evidence of residual auto-correlation, following the occurrence of an outlier. Another hidden Markov variable γ_t is introduced taking values $(1, 2, 3)$, corresponding to 'normal noise', 'jump to outlier' and 'recovery from outlier'. This led us to the following hidden Markov model for the observation process, $x_t, t = 1, \ldots, 4050$.

$$x_t = \begin{cases} \alpha_t + v_t & \text{if } \gamma_t = 1 \\ \alpha_t + z_t & \text{if } \gamma_t = 2 \\ \eta(x_{t-1} - \alpha_{t-1}) + \alpha_t + v_t & \text{if } \gamma_t = 3 \end{cases} \tag{14}$$

where $v_t \sim N(0, \tau^2)$ and z_t is exponential with rate λ. We used the results of the moving median filter to estimate values for the unknown parameters. A rough count of about 16 jumps suggested setting $p_J = 1/250$; examining Figure 6.4 shows that reasonable guesses for μ_W and σ_W^2 are $115, 000$ and $10, 000^2$ respectively. The upper residuals from the median filter lie broadly between ± 5000, suggesting $\tau^2 = 2500$. The residuals below 7500 have a mean of -24624 so we set $\lambda = 1/17500$. We compared results for $\eta = 0.5, 0.3, 0.1$. It remains to choose the parameters for the transition matrix of the hidden Markov model. We remarked in the preceding paragraph that large outliers are often followed by a drift back towards the mean. We therefore decided to allow transitions from states $1 \to 2 \to 3$ only. Since the 70 residuals that are less than -7500 occur in 16 clusters, we set the probability of moving from state $1 \to 2$ to be $1/250$. This gives the following transition matrix,

$$\begin{pmatrix} 0.995 & 0.004 & 0 \\ 0 & (1-q) & q \\ q & 0 & (1-q) \end{pmatrix}, \tag{15}$$

where the probabilities of moving from state $1 \to 2$ and $3 \to 1$ both set to q. Since the average sojourn time in an outlier state is 4, a natural choice for q is 0.5.

We note that while the GPB(2) filter could be extended to this situation, it would now involve an extra level of complexity, namely merging a four parameter density. Further, in this example we are particularly interested in estimating the time of the last jump, and this is not possible with the GPB(2) algorithm, which merges all the histories at time $t-1$.

In order to apply the particle filter to this model, we first derive the updating equations for the posterior densities of α_t corresponding to the various states of the hidden Markov model. In all cases, these are truncated normal densities (Fearnhead, 1998). The support of each particle in the filter has six elements. These are the posterior mean and variance (μ_t, σ_t^2) of α_t, the range (a_t, b_t) on which the truncated normal distribution is supported, the current state of the hidden Markov model and the time of the last jump. Since the signal is constant for long intervals, we use Method 1 with a sample size of 1000. Details of the updating formulae are omitted for brevity.

We fitted three models, corresponding to $\eta = 0.5, 0.3, 0.1$. Since the results are similar we present the results for just one case ($\eta = 0.3$). Figure 6.5 shows the original series, together with the estimated probability of a jump (smoothed over the preceding twelve steps) and the instantaneous probability of an outlier. As expected, outliers and jumps often coincide. Elsewhere, the filter appears to distinguish between the two fairly well.

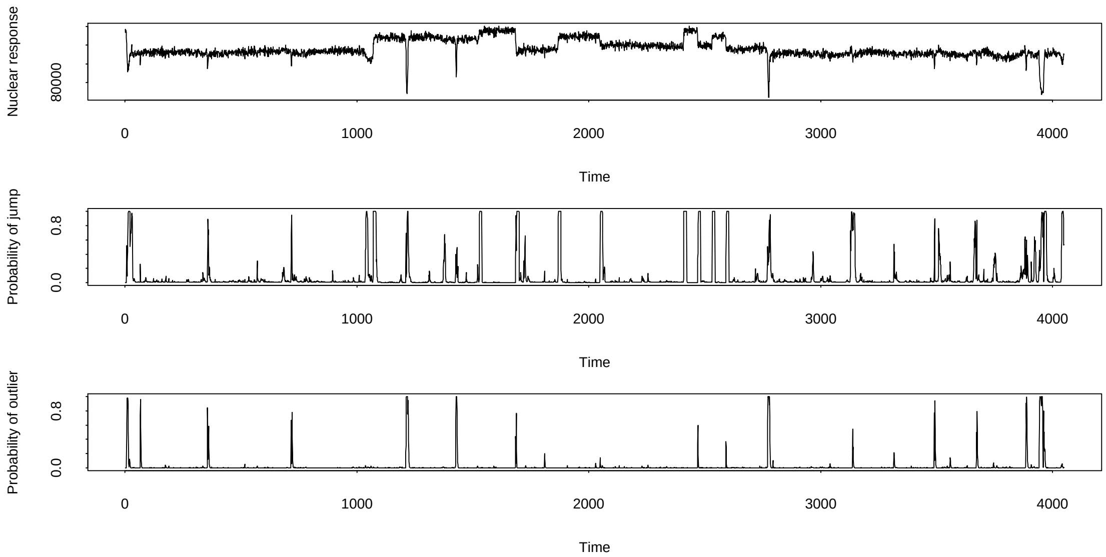

Figure 6.5: Original series (top) with estimated probability of an jump, smoothed over the preceding twelve steps (middle) and probability of an outlier (bottom)

The original aim of the analysis was to identify the jumps. One way to decide if the series has jumped is to compare the posterior probability of this event to a threshold. If this is done for the model with $\eta = 0.1$ with a threshold of 0.9 then 23 jumps are detected. If we exclude all points with a reasonable probability of being an outlier, an on-line estimate of the mean can be obtained by averaging the points since the last jump. Excluding those points which have more than a 10 % chance of being an an outlier, we obtain the online estimate in Figure 6.6

6.3 Ion channel data

Understanding the transport of ions through channels is fundamental to the understanding of a number of physiological processes. Since, despite the impressive achievements of cryoelectron microscopy, direct observation of the relevant processes remains problematic (Sansom, 1993), inferences about these processes must be drawn from sequences such as that shown in Figure 6.7. These data, obtained by the technique of patch clamp recording, represent current flow through a single ion channel in a cell membrane. The current flow clearly switches be-

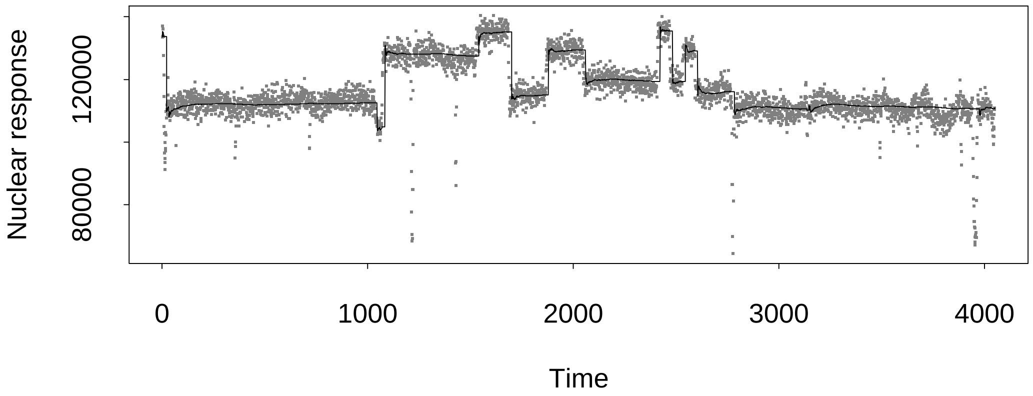

Figure 6.6: Online estimate of the underlying signal. One hundred and twenty eight outliers were identified and eliminated by the filter

tween three stages in a seemingly random manor. In fact, the highest level (around 20,000) corresponds to the ion channel being closed, and the two lower levels (around -10,000 and -30,000) correspond to two different open 'levels'.

There is an extensive statistical literature on ion channels; see Ball and Rice (1992) for an overview. Here we focus on recovering the underlying signal, since this enables us to estimate two quantities of particular interest to biophysicists: namely the mean current flow in each state and the probability of jumping from one state to another.

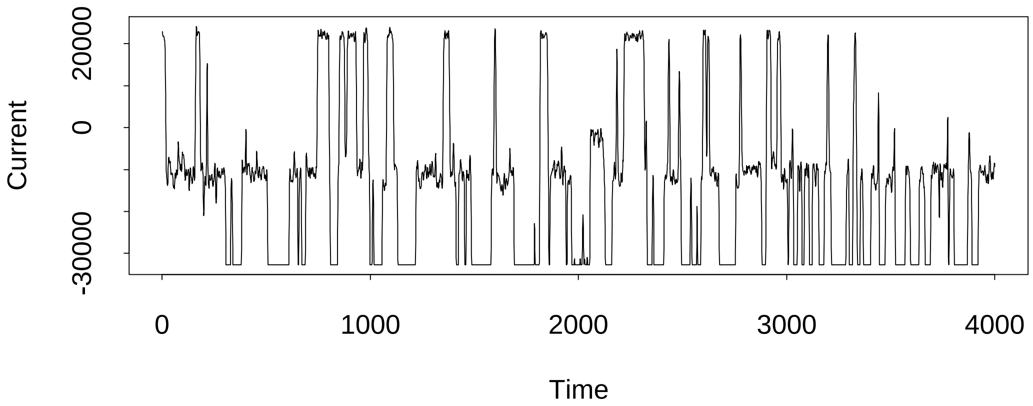

Figure 6.7: Four thousand observations from a record of length 250,000 of an NMDA-type glutamate receptor in a dentate gyrus granule cell (Colquhoun and Sigworth, 1995, p. 509). Data provided by David Colquhoun.

While the underlying signal can be estimated 'off line' (e.g. Hodgson (1999), Fredkin and Rice (1992)), such approaches are of limited applicability, since in data sets of a realistic size, the computations are intractable. We therefore apply Method 1 to recover the signal, obtaining an 'on-line' estimate of the mean and standard error of the current flow in each state, together with an estimate of the probability of jumping from one state to another. We prefer Method 1 to Method 2 since the true signal is constant for long intervals (c.f. the discussion earlier in this section).

The data we have used in Figure 6.7 had been previously smoothed using a Gaussian kernel (Colquhoun and Sigworth, 1995, p. 578) to remove outliers while retaining evidence of very brief changes in channel behaviour. Consequently it is important for the filter to detect slight divergences from the typical variation about the mean in a particular state, since these represent brief sojourns in an adjacent state.

Prior to filtering the data, we used a preceding series of 1000 points to choose our model and its parameters, henceforth referred to as the *training* data. Like Figure 6.7, the training data has three levels, with approximate means of $20,000$, $-10,000$ and $-30,000$ respectively. We first filtered the training data using a median filter with a window length of 50. This showed that the pattern of variation at the three levels is quite distinct. In the lowest level, there is virtually no variation about the mean. This is because the amplification system for the original signal saturated at this level. In both the upper levels, the variation is well described by an ARMA(2,1) model, so that

$$x_t = \zeta_i(x_{t-1} - \alpha_{t-1}) + \xi_i(x_{t-2} - \alpha_{t-2}) + \eta_i\epsilon_{t-1} + \alpha_t + \epsilon_t, \qquad i = 1, 2, \tag{16}$$

where x_t is the observed data, α_t the level of the process at time t and the ϵ_t are independent Gaussian variables.

We therefore chose a model with basic states 1–3 corresponding to the upper middle and lower levels of the data. At each time point the model can either carry on in the current state or switch to an adjacent state. If the model remains in the same state then the estimates of the state mean and variance are updated using the Kalman filter. Otherwise, the model moves into a 'jumping' state, where the observation is assumed to come from an adjacent state, but with a much larger variance. If we allow only jumps to adjacent states, then this gives rise to a seven state hidden Markov model (HMM).

Each particle of the filter maintains an estimate of the mean and variance of each of the three levels, together with the state the HMM model is currently in and the time of the last jump. When the HMM model is in states 1–3 the appropriate estimates of the mean and variance of that state are updated using appropriate model and the Kalman filter. States 4–7 of the HMM represent jumping between levels $1 \to 2$, $2 \to 3$, $3 \to 2$ and $2 \to 1$ respectively. In these HMM states, the observation is assumed to come from the level to which the model is moving, and the estimates of the mean and variance of that level are not updated.

The parameters of the transition matrix, P, were estimated by counting the number of likely transitions in the training data, and are given by (17), where the entry in the i^{th} row and j^{th} column is the probability of moving from state i to state j.

$$P = \begin{pmatrix} 0.98 & 0 & 0 & 0.02 & 0 & 0 & 0 \\ 0 & 0.96 & 0 & 0 & 0.02 & 0 & 0.02 \\ 0 & 0 & 0.98 & 0 & 0 & 0.02 & 0 \\ 0.01 & 0.95 & 0 & 0.04 & 0 & 0 & 0 \\ 0 & 0.01 & 0.95 & 0 & 0.04 & 0 & 0 \\ 0 & 0.95 & 0.01 & 0 & 0 & 0.04 & 0 \\ 0.95 & 0.01 & 0 & 0 & 0 & 0 & 0.04 \end{pmatrix}. \tag{17}$$

At each time point, α_t takes on one of three values, representing the signal in the closed and two open states of the ion channel. Denote the posterior means of these three values at time t by $\mu_{t,i}$, $i = 1, 2, 3$, and the posterior variances by $\sigma^2_{t,i}$. If the HMM model is in state 1 then $\mu_{t,1}$

and $\sigma^2_{t,1}$ are updated according to the ARMA(2,1) process (16) with parameters ζ_1, ξ_1, η_1 and $\operatorname{Var} \epsilon_t = \tau_1^2$, while the other parameters remain unchanged. Similarly, if the HMM model is in state 2 then $\mu_{t,2}$ and $\sigma^2_{t,2}$ are updated according to the ARMA(2,1) process (16) with parameters ζ_2 ξ_2, η_2 and $\operatorname{Var} \epsilon_t = \tau_2^2$, while the other parameters remain unchanged. If the HMM model is in state 3 then the observation process is assumed to be $x_t = \alpha_t + \epsilon_t$, and $\mu_{t,3}$, $\sigma^2_{t,3}$ are updated accordingly, with $\operatorname{Var} \epsilon_t = \tau_3^2$, while the other parameters remain unchanged. Finally, if the model is in one of the jumping states 4–7, then the observation process is assumed to be $x_t = \alpha_t + \epsilon_t$, where $\operatorname{Var} \epsilon_t = \tau_4^2$, and the means and variances are not updated.

It remains to choose values for the various means and variances. This was done by estimating values from the training data, after filtering it with a median filter with window width 50. This suggested the parameter estimates shown in Table 3, and initial values for the means and

Parameter	$\zeta_1 = \zeta_2$	$\xi_1 = \xi_2$	$\eta_1 = \eta_2$	τ_1	τ_2	τ_3	τ_4
Value	1.6	-0.8	0.8	150	400	50	5000

Table 3: Table of parameter values, obtained from training data, for ion channel filtering

variances of $(\mu_{1,1}, \mu_{1,2}, \mu_{1,3}) = (22000, -10000, -32700)$ and $(\sigma_{1,1}, \sigma_{1,2}, \sigma_{1,3}) = (50, 150, 1)$.

Figure 6.8 presents the results of filtering the data shown in Figure 6.7. Note in particular that many instantaneous changes, particularly between levels 3 and 2, are recovered. In three places, the signal appears to jump straight from the second of the open states to the closed state, a situation which the particles in the filter are not allowed to do, c.f. (17). However, the filter does not diverge, and yields a reasonable estimate of the signal. Of course, if desired the transition matrix could be modified to allow jumps of more than one state at a time. The lower panel of the figure shows that the filter is not sticking in the 'jumping' states 4–7, but rather only visiting these briefly (under 12% of the time), to jump between states 1–3, as was intended. From the filter, we can recover on-line estimates of the means and variances of the

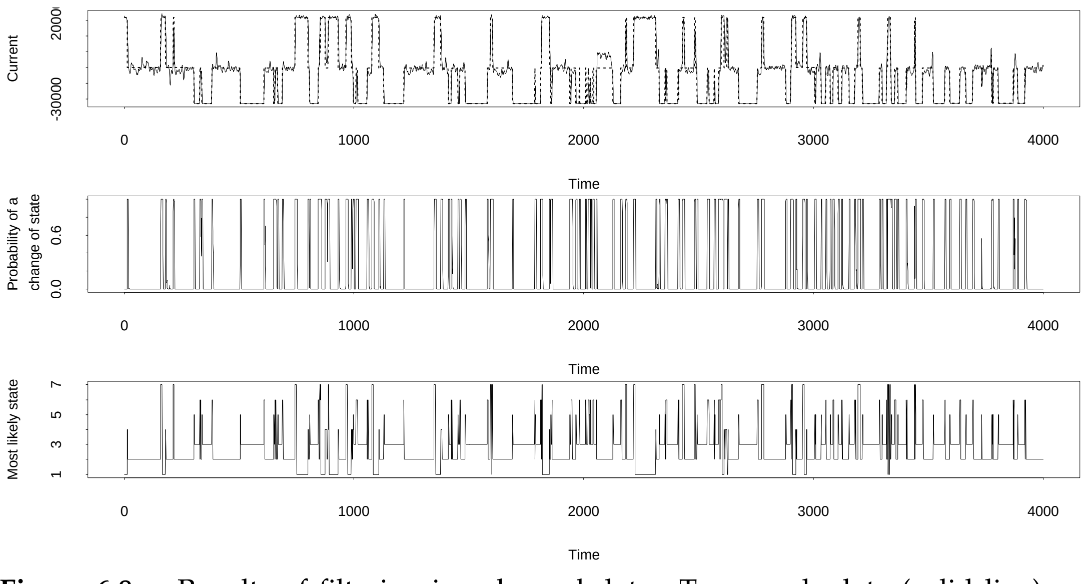

Figure 6.8: Results of filtering ion channel data. Top panel: data (solid line) and on-line estimated signal (broken line). Centre panel, probability of a jump at time t. Lower panel, most likely state at time t. Filtering carried out with Method 1 and a sample size of 100.

three levels of the signal, together with the probabilities of jumping between states. The final estimates are presented in Table 4.

Table 4 shows that the filter has obtained improved estimates of the level values of the three

Parameter	μ_1	μ_2	μ_3	σ_1	σ_2	σ_3	p_{12}	p_{23}	p_{32}	p_{21}
Prior	22000	-10000	-32768	50	150	1	0.020	0.020	0.020	0.020
Posterior	21892	-10186	-32738	31	45	0.8	0.0375	0.195	0.0272	0.006

Table 4: Comparison of means of posterior parameter distributions with the prior values.

states. Finally, Figures 6.9 and and 6.10 show the residuals. Residuals were deemed to be from

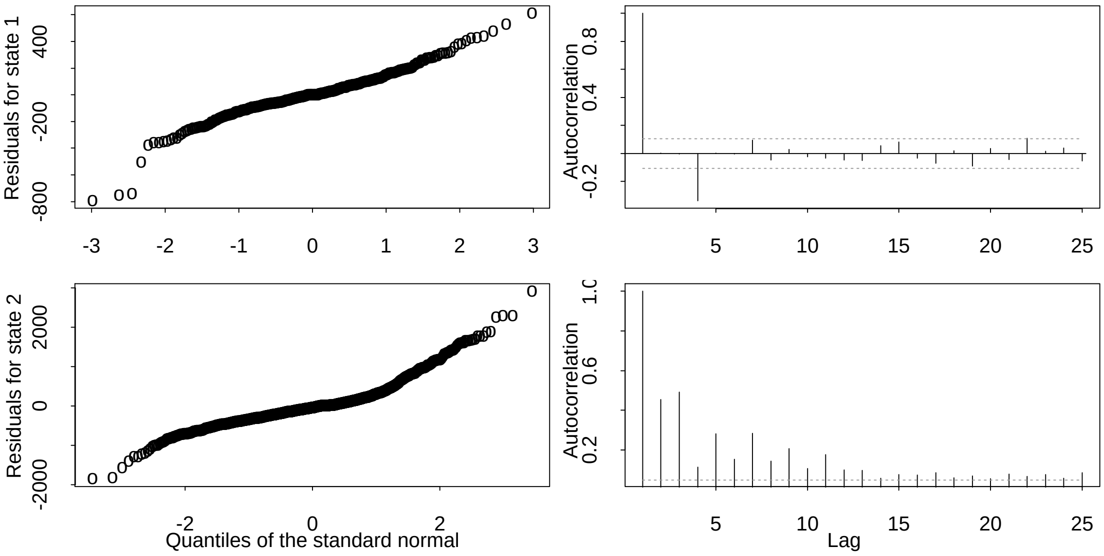

Figure 6.9: Residuals from filtering ion channel data. Left panels, normal probability plots and right panels, autocorrelation for residuals from state 1 (upper panels) and state 2 (lower panels).

a particular state if the probability of being in that state was greater than 0.9. For states 3-7, normal quantile and autocorrelation plots were produced in the usual way. For each visit to states 1 and 2, the residuals were calculated as

$$r_t = (x_t - \mu_t) - \zeta_i(x_{t-1} - \mu_{t-1}) - \xi_i(x_{t-2} - \mu_{t-2}) - \eta_i r_{t-1},$$

where μ_t is the weighted mean of α_t for each state at time t. The residuals from state 3 are not normally distributed, and are mostly zero, as expected. The large values almost always occur at the end of a sojourn in state 3. Their number could be reduced by reducing τ_3, in which case they would join the residuals from the 'jumping' states. Otherwise, the residuals are, to a reasonable approximation, normally distributed.

Turning to the auto-correlation of the residuals, it appears that the attempt to model this in states 1 and 2 has met with limited success, particularly in state 2. There is some autocorrelation in the residuals from the jumping states, as expected. This could perhaps be tackled by introducing an AR process for these.

In conclusion, we have shown how particle filters can be used successfully to sequentially filter ion channel data . Because of their extra flexibility, relative to the Kalman filter and its extensions, they can, if necessary, copy with complex, non-linear variation. Ultimately, it is hoped they could be implemented in real time.

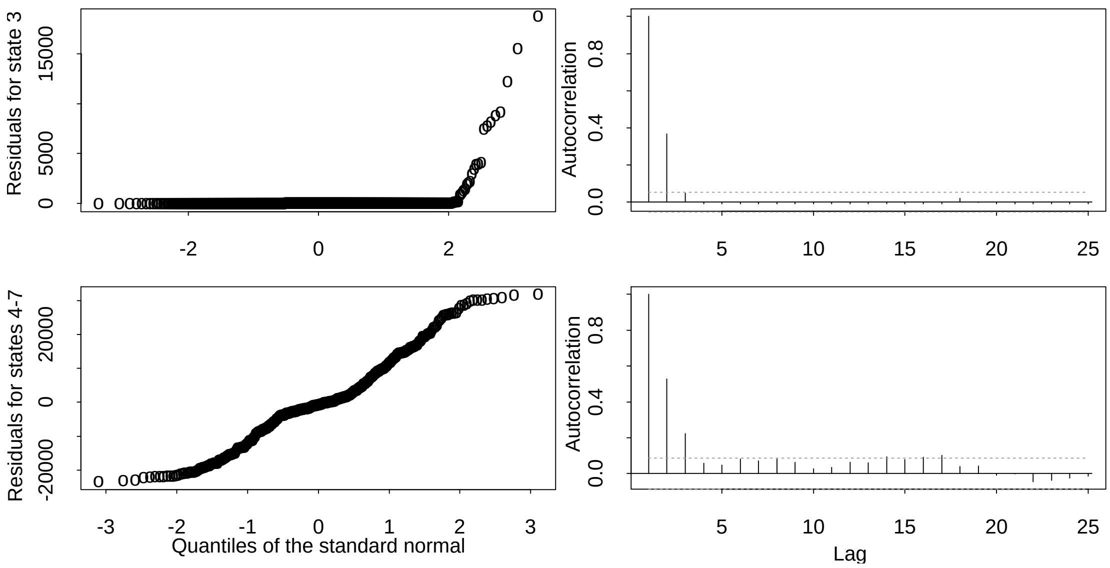

Figure 6.10: Residuals from filtering ion channel data. Left panels, normal probability plots and right panels, autocorrelation for residuals from state 3 (upper panels) and states 4-7 (lower panels).

7 Conclusion

The automation of data collection is becoming increasingly routine. There are obvious benefits in being able to process records as they arrive with a view to extracting all the useful infomation and discarding the residue. Consequently, there is a need to develop accurate, sequential filters. This need has been reflected in the recent rise in interest in particle filters in the literature. Such filters have the advantage over traditional methods, such as the Kalman filter, that they can easily cope with non-linear state spaces and observation processes together with non-Gaussian errors. However, in their simplest form, they suffer from sample impoverishment, which can lead to impractically large sample sizes being required to avoid gross biases and filter divergence. We believe that particle filters can be improved by the application of effective Monte Carlo integration techniques. Wherever possible randomness should be eliminated and replaced by analytic calculations. In particular, we advocate the use of particle filters indexed by parametric information, where possible, and the use of systematic sampling, both of which lead to substantial improvements in filter performance.

8 Acknowledgements

This research was supported by DERA grant WSS/U1172. Paul Fearnhead has an EPSRC studentship. We are grateful to Neil Gordon and David Salmond for a number of stimulating conversations, and to Bill Fitzgerald and David Colquhoun for providing the data.

9 References

Acklam, P. J. (1996) *Monte Carlo methods in state space estimation*. Master's thesis, Department of mathematics, University of Oslo.

Aidala, V. J. (1979) Kalman filter behaviour in bearings-only tracking applications. *IEEE Transactions on Aerospace and Electronic Systems*, **15**, 29–39.

Aidala, V. J. and Hammel, S. E. (1983) Utilization of modified polar coordinates for bearings-only tracking. *IEEE Transactions on Automatic Control*, **28**, 283–294.

Aidala, V. J. and Nardone, S. C. (1982) Biased estimation properties of the pseudolinear tracking filter. *IEEE Transactions on Aerospace and Electronic Systems*, **18**, 432–441.

Akashi, H. and Kumamoto, H. (1977) Random sampling approach to state estimation in switching environments. *Automatica*, **13**, 429–434.

Alspach, D. L. and Sorenson, H. W. (1972) Non-linear Bayesian estimation using Gaussian sum approximation. *IEEE Trans. Auto. Control*, **17**, 439–447.

Ball, F. G. and Rice, J. A. (1992) Stochastic models for ion channels: Introduction and bibliography. *Mathematical Biosciences*, **112**, 189–206.

Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1996) Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Am. Statist. Assoc. To appear*.

Besag, J. and Green, P. J. (1993) Spatial statistics and Bayesian computation. *J. Roy. Statist. Soc. B*, **55**, 25–27.

Blom, H. A. P. and Bar-Shalom, Y. (1988) The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, **33**, 780–783.

Bucy, R. S. (1969) Bayes theorem and digital realiasation for nonlinear filters. *Journal of Austronautical Science.*, **17**, 80–94.

Caflisch, R. E., Morokoff, W. and Owen, A. (1997) Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. *Technical report*, Stanford University, Statisitics Department.

Carpenter, J. R., Clifford, P. and Fearnhead, P. (1998) An improved particle filter for non-linear problems. *IEE proceedings-F (in press)*.

Colquhoun, D. and Sigworth, F. J. (1995) Practical Analysis of records. In *Single channel recording* (Eds B. Sakmann and E. Neher), pp. 483–588. New York: Plenum press, second edition.

Crisan, D. and Lyons, T. (1997) A particle approximation of the solution of the Kushner-Stratonovitch equation. *Technical report*, Imperial College, 180 Queen's Gate, London, SW7 2BZ, Department of Mathematics.

Doucet, A. (1998) On sequential simulation based methods for bayesian filtering. *Technical Report CUED/F-INFENG/TR.310*, University of Cambridge, Signal Processing Group, Department of Engineeing.

Fearnhead, P. (1998) *Sequential Monte-Carlo methods in filter theory*. Ph.D. thesis, Oxford University, Department of Statistics.

Fishman, G. S. (1996) *Monte Carlo—Concepts, Algorithms and Applications.* New York: Springer.

Fredkin, D. R. and Rice, J. A. (1992) Bayesian restoration of single channel patch clamp recordings. *Biometrics*, **48**, 427–448.

Gilks, W. R., Clayton, D. G., Spiegelhalter, D. J., Best, N. G., McNeil, A. J., Sharples, L. D. and Kirby, A. J. (1993) Modelling complexity : Applications of Gibbs sampling in medicine. *J. Roy. Statist. Soc. B*, **55**, 39–52.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996) *Markov Chain Monte Carlo in practice*. London: Chapman and Hall.

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

Gordon, N. J. (1993) *Bayesian methods for tracking*. Ph.D. thesis, Imperial College, University of London.

Hammersley, J. M. and Handscomb, D. C. (1964) *Monte Carlo Methods.* London: Methuen.

Handschin, J. (1970) Monte carlo techniques for prediction and filtering of nonlinear stochastic processes. *Automatica*, **6**, 555–563.

Handschin, J. E. and Mayne, D. Q. (1969) Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, **9**, 547–559.

Hodgson, M. E. A. (1999) A bayesian restoration of an ion channel signal. *Journal of the Royal Statistical Society, Series B (to appear)*.

Isard, M. and Blake, A. (1996) Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. Computer Vision*, pp. 343–356, Cambridge UK.

Jazwinski, A. H. (1973) *Stochastic processes and filtering theory*. Academic Press.

Johnson, G. W., Cohen, A. O., Ohlms, D. E. and Shier, C. W. (1983) Modified polar coordinates for ranging from Doppler and bearing measurements. In *Proc ICASSP 83 Boston Conf.*, pp. 907–910. IEEE.

Julier, S., Uhlmann, J. and Durrant-Whyte, H. F. (1998) A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic control.*, p. To appear.

Kitagawa, G. (1987) Non-Gaussian state-space modelling of non-stationary time series (with discussion). *J. Amer. Statist. Assoc.*, **82**, 1032–1063.

Kitagawa, G. (1996) Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5**, 1–25.

Kramer, S. C. and Sorenson, H. W. (1988) Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, **24**, 789–801.

Lindgren, A. G. and Gong, K. F. (1978) Position and velocity estimation via bearing observations. *IEEE Transactions on Aerospace and Electronic Systems*, **14**, 564–577.

Liu, J. S. (1996) Metropolised independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, **6**, 113–119.

Liu, J. S. and Chen, R. (1997) Sequential Monte Carlo Methods fo Dynamic Systems. *Technical report*, Stanford University, Department of Statistics.

Liu, J. S., Chen, R. and Wong, W. H. (1996) Rejection control and importance sampling. *Technical report*, Stanford University, Department of Statistics.

MacEachern, S. N., Clyde, M. A. and Liu, J. S. (1998) Sequential importance sampling for nonparametric Bayes models. *Canadian Journal of Statistics (to appear)*.

Masreliez, C. J. (1975) Approximate non-Gaussian filtering with linear state and observation relations. *IEEE Trans. Auto. Control*, **20**, 107–110.

Moon, J. R. and Stevens, C. F. (1996) An approximate linearisation approach to bearings-only tracking. *IEE Target tracking and data fusion: Colloquium digest 96/253*, pp. 8/1–8/14.

Nardone, S. C., Lindgren, A. G. and Gong, K. F. (1984) Fundamental properties and performance of conventional bearings-only target motion analysis. *IEEE Transactions on Automatic Control*, **29**, 775–787.

Neal, R. M. (1998) Annealed Importance Sampling. *Technical Report 9805*, Department of Statistics and Computing Science, University of Toronto,Toronto.

Ó Ruanaidh, J. J. K. and Fitzgerald, W. J. (1996) *Numerical Bayesion Methods Applied to Signal Processing*. New York: Springer.

Pitt, M. K. and Shephard, N. (1997) Filtering via simulation: auxiliary particle filters. *Technical report*, Nuffield College, Oxford University, Oxford OX1 1NF, UK.

Pole, A. and West, M. (1990) Efficient Bayesian learning in dynamic models. *Journal of Forecasting*, **9**, 119–136.

Rubin, D. (1987) Comment on 'The calculation of posterior distributions by data augmentation' by Tanner, M. A. and Wong W. H. *J. Am. Statist. Assoc.*, **82**, 543.

Rubin, D. B. (1988) Using the SIR algorithm to simulate posterior distributions. In *Bayesian Statistics 3* (Eds J. M. Bernado, M. H. DeGroot, D. V. Lindley and A. F. M. Smith), pp. 395–402, Oxford: Oxford University Press.

Sansom, M. S. P. (1993) Structure and function of channel-forming peptaibols. *Quarterly reviews of biophysics*, **26**, 365–421.

Silverman, B. W. (1986) *Density estimation for statistics and data analysis*. London: Chapman and Hall.

Smith, A. F. M. and Gelfand, A. E. (1992) Bayesian Statistics Without Tears: A Sampling-Resampling Perspective. *The American Statistician*, **46**, 84–88.

Smith, A. F. M. and Roberts, G. O. (1993) Bayesian computation via the Gibbs sampler and related Markov Chain Monte Carlo methods. *J. Roy. Statist. Soc. B*, **55**, 3–23.

Sorenson, H. W. (1988) Recursive estimation for nonlinear dynamic systems. In *Bayesian analysis of time series and dynamic models* (Ed. J. C. Spall), Dekker.

Tugnait, J. K. (1982) Detection and estimation for abruptly changing systems. *Automatica*, **18**, 607–615.

Weiss, H. and Moore, J. B. (1980) Improved extended Kalman filter design for passive tracking. *IEEE Transactions on Automatic Control*, **25**, 807–811.

West, M. and Harrison, P. (1997) *Bayesian Forecasting and Dynamic Models.* New York: Springer, second edition.

West, M., Harrison, P. J. and Migon, H. S. (1985) Dynamic generalised linear models and Bayesian forecasting (with discussion). *J. Am. Statist. Assoc.*, **80**, 73–97.

A Effective sample size in importance sampling

Suppose that $\pi(\theta)$ is a target density, and that $p(\theta)$ is a proposal density, with a larger support set. Let $r(\theta) = \pi(\theta)/p(\theta)$, and let $g(\theta)$ be a real valued function of θ. The importance sample estimate of $\mu = \mathbf{E}_\pi g(\theta)$ is

$$\hat{\mu} = \frac{\sum_{i=1}^n r(\theta^i) g(\theta^i)}{\sum_{i=1}^n r(\theta^i)},$$

where $\{\theta^i\}_{i=1,\dots,n}$ is a sample from $p(\theta)$. It can be verified that the large sample variance of this ratio is

$$\mathrm{Var}(\hat{\mu}) = \frac{1}{n}\mathbf{E}_p(g(\theta)-\mu)^2 r(\theta)^2 + O(n^{-2}),$$

where the subscripts π and p indicate which density is being used to calculate the expectation. Thus, for large n, the effective sample size for estimating μ is given by

$$n_e(g) = \frac{n\mathbf{E}_\pi(g(\theta)-\mu)^2}{\mathbf{E}_p(g(\theta)-\mu)^2 r(\theta)^2}.$$

This is to be compared with the expression for the effective sample size proposed by Liu (1996), namely the estimated value of

$$\frac{n}{1+\ \mathrm{Var}_p r(\theta)}.$$

We will compare these values for the simple case of estimating $\mu = \mathbf{E}_\pi\theta$.

LEMMA A.1 *Let $g(\theta) = \theta$ and $p(\theta)$ be the standard normal density.*

i) If

$$\pi(\theta) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{\theta^2}{2\sigma^2}\right\},$$

where $\sigma^2 < 2$, then as $\sigma^2 \to 2$,

$$n_e(g)\ \frac{1+\ \mathit{Var}_p r(\theta)}{n} \to 0.$$

ii) If

$$\pi(\theta) = \begin{cases} c\exp\{-\theta^2/2\} & |\theta| > \epsilon \\ a & |\theta| \le \epsilon \end{cases},$$

where $a < 1/(2\epsilon)$ and ϵ are positive parameters of the above distribution, and c is a normalising constant, then as $a \to \infty$ and $\epsilon \to 0$, such that $a\epsilon$ remains fixed,

$$n_e(g)\ \frac{1+\ \mathit{Var}_p r(\theta)}{n} \to \infty.$$

PROOF:
In both cases $\mu = 0$. Further $1 +\ \mathrm{Var}_p r(\theta) = \mathbf{E}_p r(\theta)^2$ as $\mathbf{E}_p r(\theta) = 1$, by definition of $r(\theta)$. Consider the two case separately:

i) In this case $r(\theta)^2 = \exp\{-\theta^2(1/\sigma^2-1)\}/\sigma^2$, and $\mathbf{E}_\pi(g(\theta)-\mu)^2 = \sigma^2$. So, for $\sigma^2 < 2$,

$$\begin{aligned}\mathbf{E}_p(\theta^2 r(\theta)^2) &= \int_{-\infty}^{\infty}\theta^2\exp\{-\theta^2(1/\sigma^2-1/2)\}/(\sqrt{2\pi}\sigma^2)\mathrm{d}\theta \\ &= (2/\sigma^2-1)^{-1}\mathbf{E}_p r(\theta)^2.\end{aligned}$$

Therefore

$$\frac{\sigma^2[1+\ \mathrm{Var}_p r(\theta)]}{\mathbf{E}_p(g(\theta)-\mu)^2 r(\theta)^2} = (2-\sigma^2) \to 0,$$

as $\sigma^2 \to 2$.

ii) For this case $r(\theta)^2 = 2\pi c^2$ if $|\theta| > \epsilon$, and $r(\theta)^2 = 2\pi a^2 \exp\{\theta^2\}$ otherwise. So, if Z is a standard normal random variable, then

$$\begin{aligned} \mathbf{E}_p r(\theta)^2 &= 4\pi c^2 \mathbf{P}(Z > \epsilon) + \sqrt{2\pi}\int_{-\epsilon}^{\epsilon} a^2 \exp(\theta^2/2)\mathrm{d}\theta \\ &\geq 4\pi c^2 \mathbf{P}(Z > \epsilon) + 2\sqrt{2\pi}\epsilon a^2, \end{aligned}$$

and

$$\begin{aligned} \mathbf{E}_p \theta^2 r(\theta)^2 &= 4\pi c^2 \int_{\epsilon}^{\infty} \theta^2 p(\theta)\mathrm{d}\theta + \sqrt{2\pi}\int_{-\epsilon}^{\epsilon} \theta^2 a^2 \exp(\theta^2/2)\mathrm{d}\theta \\ &\leq 2\pi c^2 + 2\sqrt{2\pi}\epsilon^3 a^2 \exp\{\epsilon^2/2\}, \end{aligned}$$

where since c is the normalising constant,

$$1 = 2a\epsilon + 2c\int_{\epsilon}^{\infty} \exp\{-\theta^2/2\}\mathrm{d}\theta.$$

Finally,

$$\mathbf{E}_\pi(\theta-\mu)^2 = 2c\int_{\epsilon}^{\infty} \theta^2 \exp(-\theta^2/2)\mathrm{d}\theta + 2a\epsilon^3/3.$$

Thus, fixing $a\epsilon$ at say $\delta < 1/2$, as $a \to \infty$ and $\epsilon \to 0$, we have $\mathbf{E}_p r(\theta)^2 \to \infty$, $\mathbf{E}_p \theta^2 r(\theta)^2 \to (1-2\delta)^2$ and $\mathbf{E}_\pi(\theta-\mu)^2 \to (1-2\delta)$. The required result follows. ■

This shows that for certain $p(\theta)$, $\pi(\theta)$ and $g(\theta)$ the approximated effective sample size can be infinitely wrong.

It should be noted that $n_e(g)$, the actual effective sample with importance sampling for any function of interest can be estimated quite simply by the moment estimator

$$\hat{n}_e(g) = \frac{n\sum_{i=1}^{n}(g(\theta^i)-\hat{\mu})^2 r(\theta^i)}{\sum_{i=1}^{n}(g(\theta^i)-\hat{\mu})^2 r(\theta^i)^2}.$$

This requires very little more effort than the calculation of the empirical variance of the set of weights $\{r(\theta^i)\}$.