# An Improved Particle Filter for Non-linear Problems

James Carpenter Peter Clifford† Paul Fearnhead

† *Address for correspondence:* Department of Statistics, University of Oxford,
1 South Parks Road, Oxford, OX1 3TG, UK.

**Abstract**

The Kalman filter provides an effective solution to the linear-Gaussian filtering problem. However, where there is nonlinearity, either in the model specification or the observation process, other methods are required. We consider methods known generically as *particle filters*, which include the condensation algorithm and the Bayesian bootstrap or sampling importance resampling (SIR) filter. These filters represent the posterior distribution of the state variables by a system of particles which evolves and adapts recursively as new information becomes available. In practice, large numbers of particles may be required to provide adequate approximations and for certain applications, after a sequence of updates, the particle system will often collapse to a single point. We introduce a method of monitoring the efficiency of these filters, which provides a simple quantitative assessment of sample impoverishment and show how to construct improved particle filters which are both structurally efficient, in terms of preventing the collapse of the particle system and computationally efficient in their implementation. We illustrate with the classic bearings-only tracking problem.

*Keywords:* Kalman filter, Condensation algorithm, Bayesian bootstrap filter, Sampling importance resampling (SIR) filter, Sequential estimation, Markov Chain Monte Carlo (MCMC), Importance resampling.

## 1 Introduction

The Bayesian approach to dynamic state estimation problems involves the construction of the probability density function (PDF) of the current state of an evolving system, given the accumulated observation history. For linear Gaussian models where the PDF can be summarised by means and covariances, the calculation is carried out in terms of the familiar updating equations of the Kalman filter. In general, for nonlinear, non-Gaussian models, there is no simple way to proceed. Two difficulties must be resolved: how to represent a general PDF using finite computer storage and how to perform the integrations involved in updating the PDF when new data are acquired.

Several approximate methods have been proposed. These include the extended Kalman filter [1, 2], the Gaussian sum filter [3], approximating the first two moments of the PDF [4, 5] and numerical integration over a grid of points in the state space [6–10] . However, none of these methods can be applied automatically. Typically, they have to be tuned to take account of features of each specific problem. For example, in grid-based methods the number and location of the grid points has to be decided upon, usually by a process of trial and error. Furthermore, updating the distribution of the state of the system as new data arrive usually entails a formidable computational overhead.

There is now a substantial literature concerned with simulation based filters in which the required PDF is represented by a scatter of particles which propagate through the state space [9,11–19]. The propagation and adaptation rules are chosen so that the combined weight of particles in a particular region will approximate the integral of the PDF over the region. Such filters have been variously described as Bayesian bootstrap, condensation, Monte-Carlo and Metropolis-Hastings importance resampling filters. For the purposes of this paper we adopt the term Monte Carlo particle filter or particle filter for short. Gordon, Salmond and Smith [14] demonstrate the effectiveness of a simple algorithm for particle evolution for various nonlinear filtering applications. Their method has become known as the Sampling Importance Resampling (SIR) filter or, more commonly in the engineering literature, the Bayesian bootstrap filter.

The standard SIR filter is vulnerable to *sample impoverishment* [17, 20–22], so that the particle distribution gives a poor approximation of the required PDF. In extreme cases, after a sequence of updates the particle system can collapse to a single point. In less extreme cases, although several particles may survive, there is so much internal correlation that summary statistics behave as if they are derived from a substantially smaller sample. To compensate, large numbers of particles are required in realistic problems.

In this paper we show how sample impoverishment can be quantified. We introduce modifications which we demonstrate to have superior performance to the SIR filter both in terms of combating sample impoverishment and in computational cost.

In section 2 we give a brief review of Bayesian filtering theory. In Section 3 we reformulate the SIR filter as an evolving random measure and show that variance reduction techniques from the theory of Monte-Carlo integration [23] can be applied. Section 4 introduces the modified particle algorithm, and we note that the computational complexity of each update calculation can be reduced from $O(N \log N)$ to $O(N)$ where $N$ is the number of particles. Section 5 demonstrates how to quantify sample impoverishment and Section 6 contains an application of the new algorithm to the classic bearings-only tracking problem. Finally Section 7 presents some conclusions.

## 2 Bayesian Filtering

Following Gordon, Salmond and Smith [14] we represent the state vector at time $k$ by $x_k \in \mathbb{R}^n$, which satisfies

$$x_{k+1} = f_k(x_k, w_k)$$

where $f_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the system transition function and $w_k$ is a noise term whose known distribution is independent of time. At each discrete time point an observation $y_k \in \mathbb{R}^p$ is obtained, related to the state vector by

$$y_k = h_k(x_k, v_k)$$

where $h_k : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^p$ is the measurement function and $v_k \in \mathbb{R}^r$ is another noise term whose known distribution is independent of both the system noise and time. We write $D_k$ for $(y_1, ..., y_k)$, the available information at time $k$, and assume the PDF of $x_1$, the initial state of the system, is known so that $p(x_1|D_0) = p(x_1)$. We then wish to obtain the PDFs of $p(x_k|D_k) : k \geq 2$, which are given by the 3 equations:

$$p(x_k|D_{k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|D_{k-1})\, dx_{k-1} \tag{1}$$

and

$$p(x_k|D_k) = \frac{p(y_k|x_k)p(x_k|D_{k-1})}{p(y_k|D_{k-1})}, \tag{2}$$

where

$$p(y_k|D_{k-1}) = \int p(y_k|x_k)p(x_k|D_{k-1})\,dx_k. \tag{3}$$

The basic SIR algorithm, which provides approximate solutions to (1–3) is given by Gordon, Salmond and Smith [14]. See also Section 3.1 below.

# 3 Random measures

Particle filters work by providing a Monte Carlo approximation to the PDF which can be easily updated to incorporate new information as it arrives. The Monte Carlo approximation to a PDF $p(x_k)$ at time $k$ consists of a set of random nodes in the state space $(s_k^i)_{i=1\dots N}$, termed the *support*, and a set of associated weights $(m_k^i)_{i=1\dots N}$ summing to 1. The support and the weights together form a *random measure*.

The objective is to choose a measure so that

$$\sum_{i=1}^{N} g(s_k^i)m_k^i \approx \int g(x_k)p(x_k)\,dx \tag{4}$$

for typical functions $g$ of the state space. This is an approximation in the sense that the left-hand side of (4) converges (in probability) to the right-hand side as $N \to \infty$ [24].

A simple example of a random measure is obtained by sampling values $(s_k^i)_{i=1\dots N}$ independently from $p(x_k)$, and attaching equal weights $m_k^i = N^{-1}; i = 1,\dots,N$ to the values. The left-hand side of (4) is then the sample average $\sum_{i=1}^{N} g(s_k^i)/N$. Importance sampling [25] generalises this by sampling $(s_k^i)_{i=1\dots N}$ from an importance PDF $f(x_k)$ and attaching importance weights $m_k^i = Ap(s_k^i)/f(s_k^i)$, where $A^{-1} = \sum_{i=1}^{N} p(s_k^i)/f(s_k^i)$.

More sophisticated Monte Carlo integration techniques [23] are also available. Stratified sampling is of particular relevance. Suppose that a PDF $p(x)$ is made up of contributions from $N$ distinct subpopulations or strata, so that $p(x)$ is a mixture of the form

$$p(x) = \sum_{i=1}^{N} \beta_i p_i(x),$$

where each $p_i(x)$ is a PDF and $\sum_{i=1}^{N} \beta_i = 1$. Sampling theory [26] tells us that a population quantity $\int g(x)p(x)dx$ can be estimated efficiently by sampling a fixed number $M_i$ from each of the strata, with $M_1 + \dots + M_N = N$. The greatest efficiency is attained with the Neyman allocation $M_i \propto \beta_i\sigma_i$, where $\sigma_i^2$ is the variance of $g(x)$ in the $i$th stratum. In practice, either because the variances are unknown or because a number of different functions are to be monitored, the proportional allocation $N_i \propto \beta_i$ is frequently used. Except in certain degenerate cases, this proportional allocation can be shown to be more efficient than simple random sampling from $p(x)$ [26].

A random measure $(\tilde{s}^i, m^i)_{i=1\ldots N}$ which approximates $p(x)$ can be converted, by resampling, into an equally weighted measure which approximates a simple random sample from $p(x)$. Resampling consists of sampling $(s^1, \ldots, s^N)$ with replacement from $(\tilde{s}^i, m^i)_{i=1\ldots N}$, *i.e.* the discrete distribution with support points $(\tilde{s}^i)$ and probabilities $(m^i)$. This leads to a new random measure $(s^i, N^{-1})_{i=1\ldots N}$ where now the weights are equal but, typically, there are fewer distinct points in the support. Resampling plays a important role in the SIR filter but we will show that improved approximations are obtained by using the weighted measure before resampling rather than resampling and then using the unweighted measure. Intuitively, this is not surprising because we would expect a set of weighted sample points to carry more information than an equal number of unweighted points.

### 3.1 The standard SIR algorithm as a random measure

The basic SIR algorithm given by Gordon Salmond and Smith [14] is as follows.

***Initialisation*** Begin by simulating a sample $(s_1^i)_{i=1\ldots N}$ from $p(x_1)$. In other words, start from a random measure with equal weight on each of the $N$ sample values.

***Preliminaries (step k)*** Assume that we have an equally weighted random measure $(s_{k-1}^i, N^{-1})_{i=1\ldots N}$, approximating $p(x_{k-1}|D_{k-1})$.

***Prediction*** Estimate the density $p(x_k|D_k)$, up to a normalising constant $K$, by the mixture

$$p(x_k|D_k) = K \sum_{i=1}^{N} p(x_k|s_{k-1}^i)p(y_k|x_k). \tag{5}$$

Take exactly one sample point from each of the $N$ strata, by generating support points $\tilde{s}_k^i = f_{k-1}(s_{k-1}^i, w_{k-1}^i)$ from the system model, with importance weights

$$m_k^i = \frac{p(y_k|\tilde{s}_k^i)}{\sum_{j=1}^{N} p(y_k|\tilde{s}_k^j)}, \tag{6}$$

***Update*** Resample from the random measure $(\tilde{s}_k^i, m_k^i)_{i=1\ldots N}$ to obtain an equally weighted random measure $(s_k^i, N^{-1})_{i=1\ldots N}$. In other words, sample $N$ times, independently with replacement, from the set $(\tilde{s}_k^i)_{i=1\ldots N}$, with probabilities $(m_k^i)_{i=1\ldots N}$.

A rapid algorithm for the update step, is given in Subsection 4.1.

### 3.2 Variance reduction at the update stage

The analysis in this section provides the motivation for the improved filter discussed below. Suppose that we wish to estimate

$$\theta_k = \int p(x_k|D_k)g(x_k)dx_k,$$

the mean value of some function $g$ of the state of the system at time $k$. Using the resampled values from the SIR filter, this can be estimated by

$$\hat{\theta}_k = N^{-1} \sum_{i=1}^{N} g(s_k^i).$$

However, an unnecessary element of random noise is introduced by this approach. Suppose $(\tilde{s}_k^i)$ is a sample from $p(x_k|D_{k-1})$ then the quantity $\theta_k$ can be estimated more precisely by

$$\tilde{\theta}_k = \sum_{i=1}^{N} m_k^i g(\tilde{s}_k^i), \tag{7}$$

where $m_k^i$ is given in (6). To see this, we note that $\hat{\theta}_k$ can be written as

$$\hat{\theta}_k = N^{-1} \sum_{i=1}^{N} Z_i g(\tilde{s}_k^i)$$

where $(Z_1, \ldots, Z_N)$ is a multinomial distributed vector with probabilities $(m_k^i)$. The quantity $\tilde{\theta}_k$ is therefore the expected value of $\hat{\theta}_k$.

Calculating variances we find that, for $N$ large,

$$\text{Var}\, \hat{\theta}_k \approx \text{Var}\, \tilde{\theta}_k + \frac{1}{N} \text{Var}\, g(x_k), \tag{8}$$

where the last term is the variance that $\tilde{\theta}_k$ would have if $(s_k^i)_{i=1\ldots N}$ was a simple random sample of size $N$ from $p(x_k|D_k)$. Details of the derivation are omitted for brevity. It follows that the variance of $\tilde{\theta}_k$ is always smaller than that of $\hat{\theta}_k$. In fact when the observation $y_k$ is not informative, i.e. $m_k^i = N^{-1}$ then the variance of $\hat{\theta}_k$ is effectively double that of $\tilde{\theta}_k$.

## 4 Efficient Particle Filters

As a refinement of the SIR filter it has been suggested [20] that a larger number of values, say 10, should be sampled from each stratum. At the resampling stage, a sample of size $N$ is then selected from the $10 \times N$ predicted values, in order to restore the size of the support set to $N$. It has also been suggested [22] that a simple random sample should be drawn from the mixture distribution (5). However sampling theory indicates that greater accuracy can be achieved by stratified sampling.

We can write $p(x_k|D_k)$ from (5) in the form

$$p(x_k) = \sum_{i=1}^{N} \beta_i p_i(x_k)$$

where

$$\beta_i = \frac{\int p(x_k|s_{k-1}^i) p(y_k|x_k)\, dx_k}{\sum_{i=1}^{N} \int p(x_k|s_{k-1}^i) p(y_k|x_k)\, dx_k}$$

and

$$p_i(x_k) = \frac{p(x_k|s_{k-1}^i)p(y_k|x_k)}{\int p(x_k|s_{k-1}^i)p(y_k|x_k)\,dx_k}.$$

In this paper we will only consider stratification by the proportional allocation. Ideally we should take $M_i = N\beta_i$ but in practice these quantities are unlikely to be exact integers. We can however arrange that $M_i$ are integer variables with small variances and the correct expected value. Our Algorithm 2 in Section 4.1 achieves this while ensuring that $|M_i - N\beta_i|$ is always less than 1. Another simple suggestion [27] is to take $M_i$ to be the integer part of $N\beta_i$, and then to add 1 randomly with probability equal to the fractional part of $N\beta_i$. A disadvantage of this method is that the population of particles will fluctuate in size (although it will never die out completely).

In practice, the quantities $\beta_i$ and $p_i(x_k)$ may be difficult to deal with and importance sampling is necessary. Combining the preceding steps the proposed particle filter is as follows:

***Initialisation*** Start from a random measure with $N$ support points, possibly obtained by stratified sampling, which approximates the PDF $p(x_1)$.

***Preliminaries (step k)*** Assume that we have a random measure $(s_{k-1}^i, m_{k-1}^i)_{i=1\ldots N}$, approximating $p(x_{k-1}|D_{k-1})$.

***Prediction*** Estimate the density $p(x_k|D_k)$, up to a normalising constant $K$, by

$$p(x_k|D_k) = K\sum_{i=1}^{N} m_{k-1}^i p(x_k|s_{k-1}^i)p(y_k|x_k). \tag{9}$$

Construct an importance PDF

$$\hat{p}(x_k) = \sum_{i=1}^{N} \hat{\beta}_i \hat{p}_i(x_k)$$

Take a stratified sample from this density using Algorithm 2, with $M_i$ sample points in the $i$th category, where $M_i$ has expected value $N\hat{\beta}_i$.

***Update*** For each $i$, sample $M_i$ support points $(s_k^j)$ from $p_i(x_k)$, with importance weights given by

$$m_k^j \propto \frac{m_{k-1}^i p(s_k^j|s_{k-1}^i)p(y_k|s_k^j)}{\hat{\beta}_i \hat{p}_i(s_k^j)} \quad \text{for} \quad \sum_{\ell=1}^{i-1} M_\ell < j \le \sum_{\ell=1}^{i} M_\ell.$$

The updated random measure is then given by $(s_k^j, m_k^j)_{j=1\ldots N}$, where the weights are scaled to sum to 1.

Properties of the updated state distribution can be estimated using the random measure as in (7). If an approximate sample from the state distribution is required it can be obtained by simple random sampling from the random measure as described in Section 3. Note that once the stratification numbers have been calculated, there is only one sampling operation at each update. Carrying forward the weights $(m_{k-1})$ at the update step, eliminates the resampling phase of the standard SIR filter. Construction of the importance PDF is necessarily problem specific. We work through an example in Section 6.

### 4.1 Reducing the computational complexity of particle filters

Sampling of $N$ values from a discrete distribution $(s_i, m_i)_{i=1\ldots N}$, can be carried out by simulating standard uniform variables $(u_i)_{i=1\ldots N}$ and then using binary search to find the value $j$, and hence $x_j$, corresponding to

$$Q_{j-1} < u_i \leq Q_j,$$

where $Q_j = \sum_{\ell=0}^{j} m_\ell$ and $Q_0 = 0$.

Binary search is commonly used to implement the updating stage of the SIR algorithm. However it is not efficient. To obtain a sample of size $N$, by this means, takes $O(N \log N)$ calculations; the $\log(N)$ term arises from the binary search. A more efficient method is to simulate $N + 1$ exponentially distributed variables $t_0, \ldots, t_N$, using $t_i = -\log(u_i)$, calculate the running totals $T_j = \sum_{\ell=0}^{j} t_\ell$, and then merge $(T_j)$ and $(Q_j)$. The algorithm is based on the well known method of simulating order statistics [28].

**Algorithm 1** $O(N)$ algorithm for the SIR filter

```
$i = 0; j = 1$
do while $i < N$
    if $Q_j T_N > T_i$ then
        $i = i + 1$; output $s_j$
    else
        $j = j + 1$
    end if
end do
```

For precise proportional stratification, the objective is to ensure that the number of points in the $i$th category is as close as possible to $N\beta_i$. Label the categories $s_i = i$. The output will consist of $N$ category labels, with the property that the expected number of labels of category $i$ will be equal to $N\beta_i$ and the actual number will differ from the expected number by no more than 1.

**Algorithm 2** $O(N)$ algorithm for stratification

```
$T =$ unif$(0, 1)/N; j = 1; Q = 0; i = 0$
do while $T < 1$
    if $Q > T$ then
        $T = T + 1/N$; output $s_i$
    else
        pick $k$ in $\{j, \ldots, N\}$
        $i = s_k$
        $Q = Q + \beta_i$
        switch $(s_k, \beta_k)$   with   $(s_j, \beta_j)$
        $j = j + 1$
    end if
end do
```

## 5 A diagnostic for sample impoverishment

All the particle filters we compare in Section 6 are capable of approximating the posterior distributions of the state variables in a statistical sense. They differ in terms of the accuracy with which properties of the state distribution can be estimated. For the purposes of comparison, the effective sample size is an obvious quantity to compute. This is the sample size that would be required for a simple random sample from the target PDF to achieve the same estimating precision as the particle filter. Since some properties of the state distribution may be estimated well, and some poorly, the effective sample size will depend on what is being estimated.

Suppose that the property to be estimated is

$$\theta_k = \int g(x_k) p(x_k|D_k) dx_k,$$

and let

$$z_k = \sum_{i=1}^{N} m_k^i g(\tilde{s}_k^i), \quad v_k = \sum_{i=1}^{N} m_k^i g^2(\tilde{s}_k^i) - z_k^2$$

be the filter estimates of $\theta$ and $\sigma^2$, the variance of $g(x_k)$ given $D_k$ as in Equation (8). Note that, if $\theta$ is estimated by using the average value of $g(x_k)$ in a simple random sample of size $N^*$ from $p(x_k|D_k)$, the estimate will have a variance of $\sigma^2/N^*$.

To evaluate the effective sample size, we use a technique borrowed from classical ‘analysis of variance’ in statistics.

1. Run the filter independently $M$ times, obtaining $M$ independent replicates, each based on $N$ particles.
2. For each replicate, at step $k$, calculate $z_k^j$ and $v_k^j, j = 1, \ldots, M$.
3. Calculate $\bar{z}_k$ and $\bar{v}_k$, the average values over the $M$ replicates.
4. The *effective sample size* is then $M\bar{v}_k / \sum_{j=1}^{M}(z_k^j - \bar{z}_k)^2$.

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 $N^*$, *i.e.*

$$M^{-1} \sum_{j=1}^{M} (z_k^j - \bar{z}_k)^2 = \frac{\sigma^2}{N^*} \approx \frac{\bar{v}_k}{N^*}.$$

The effective sample size is then obtained by solving for $N^*$.

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. An example of such a problem is given in Section 6.

The integrated correlation time in Markov chain Monte Carlo (MCMC) calculations in non-dynamic problems [29] and the effective sample size play similar roles. Neither of these diagnostics checks to see whether there is convergence to the right distribution. A noisy biased filter may have a large effective sample size but the sample will 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 Bearings-only tracking

In this example an object moves in the $(\nu, \eta)$ plane according to a second order model

$$x_k = \Phi x_{k-1} + \Gamma w_k, \tag{10}$$

where $x_k = (\nu, \dot{\nu}, \eta, \dot{\eta})_k^T, w_k = (w_\nu, w_\eta)_k^T$,

$$\Phi = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \Gamma = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}. \tag{11}$$

The system fluctuations $w_k = (w_\nu, w_\eta)_k^T$ are independent zero-mean Gaussian white noise. The model essentially assumes that the velocity evolves like Brownian motion. The 'leap-frog' discretisation is slightly non-standard. The state variables $\dot{\nu}_k$ and $\dot{\eta}_k$ are the velocities at time $k - 1/2$. Positions are updated by using a midpoint approximation to the integrated velocities. Velocities at step $k$ would be approximated by $(\dot{\nu}_k + \dot{\nu}_{k+1})/2$ and $(\dot{\eta}_k + \dot{\eta}_{k+1})/2$. There is an equivalent formulation of the leap-frog algorithm in which $(\dot{\nu}_k, \dot{\eta}_k)$ are the velocities at time $k + 1/2$ [22, 30]. The matrix $\Gamma$ is then modified in the obvious way. We will use parameters which are compatible with an example considered by Gordon, Salmond and Smith [14] who use a different integration scheme.

The observations are a sequence of bearings:

$$y_k = \tan^{-1}(\eta_k/\nu_k) + v_k$$

The initial state of the object is $x_1 = (-0.05, 0.001, 0.7, -0.055)^T$. The system noise variables $w_\nu, w_\eta$ have variance $\tau^2 = 0.001^2$ and the observation error $v_k$ has variance $\sigma^2 = 0.005^2$. We consider an observed trajectory which passes close by the observer who is fixed at the origin. See Figure 1. We adopt the same prior distribution of the starting configuration as Gordon, Salmond and Smith [14].

To construct the importance PDF we consider the $i$th component of (9)

$$K m_{k-1}^i p(x|s) p(y|x) \tag{12}$$

where we have replaced $s_{k-1}^i$ by $s$, $y_k$ by $y$ and $x_k$ by $x$ for notational convenience. Conditional on $s$ the density of $x$ depends only on $(\nu, \eta)$. Converting $(\nu, \eta)$ and $(\mathrm{E}(\nu|s), \mathrm{E}(\eta|s))$ to polar coordinates as

$(r, \theta)$ and $(\rho, \alpha)$, respectively, it follows that (12) is proportional to

$$\begin{aligned} & m_{k-1}^i r e^{-(r^2+\rho^2-2r\rho\cos(\theta-\alpha))/(2\tau^2)-(y-\theta)^2/(2\sigma^2)} \\ = \; & m_{k-1}^i r e^{-(r-\rho\cos(\theta-\alpha))^2/(2\tau^2)-\rho^2\sin^2(\theta-\alpha)/(2\tau^2)-(y-\theta)^2/(2\sigma^2)}. \end{aligned}$$

Our importance PDF is obtained by replacing $\sin(\theta - \alpha)$ by $\theta - \alpha$, giving the PDF

$$\hat{\beta}_i f(\theta) f(r|\theta)$$

where

$$f(r|\theta) \propto r e^{-(r-\rho\cos(\theta-\alpha))^2/(2\tau^2)}$$

and $f(\theta)$ is the Gaussian PDF with mean $(\sigma^2\alpha\rho^2 + y\tau^2)/(\rho^2\sigma^2 + \tau^2)$ and variance $\sigma^2\tau^2/(\rho^2\sigma^2 + \tau^2)$ and the normalising constants have been absorbed into $\hat{\beta}_i$ with $\sum_{i=1}^N \hat{\beta}_i = 1$. Both $f(\theta)$ and $f(r|\theta)$ can be sampled directly using standard methods.

All the filters were initialised by taking samples of size $N = 5000$ from $p(x_2|D_2)$, obtained using the Metropolis Hastings Algorithm as described in [30]. The distribution of these samples was checked against a numerical evaluation of $p(x_2|D_2)$ and found to agree closely. The samples were also checked for independence by computing auto-correlations during the Metropolis Hastings simulations. The effective sample sizes were calculated using $M = 1000$ replicates. Each of the different filtering methods were successively initialised from each of the $M$ starting configurations

The results, for various methods, are presented in Table 1. We calculate effective sample sizes for the filter estimates of the mean range and the mean bearing at each step. In all cases, there is greater sample impoverishment for the range calculations than for the bearings. The ‘Improved Reweighted’ filter is an implementation of the particle filter described in the paper, with precisely stratified sampling as in Algorithm 2. The ‘Multinomial Reweighted’ filter carries weights forward as in the improved filter, but samples the strata multinomially using Algorithm 1 (with $m_i = \beta_i$). The ‘Two Stage’ algorithm [22] samples strata multinomially, but resamples after each step, to obtain equally weighted particles. The ‘Standard SIR’ [14] is included for comparison.

Note that our implementation of the basic SIR algorithm (Subsection 3.1 ) does not incorporate the *ad hoc* modifications of added jitter and prior editing suggested by Gordon, Salmond and Smith [14].

# 7 Conclusions

We have shown how to reduce the computational cost of implementing particle filters. We have proposed an improved particle filter and demonstrated its superior performance. While the filter is more complicated to implement and, unlike the standard SIR filter, needs to be tailored to the problem in hand, the computational gains are substantial. Further, we have introduced a diagnostic for sampling inefficiency which allows us to compare the performance of various Monte Carlo filters. We advocate the use of this diagnostic as a general tool in the analysis of sequential Monte Carlo algorithms.

All of the filters we have discussed suffer from substantial sample impoverishment. In principle this could be monitored using our diagnostic and compensated for dynamically, by adjusting the number of particles at critical stages. We believe that there is scope for even greater variance reduction by the use of more efficient Monte Carlo integration techniques.

## 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.

## References

[1] Jazwinski, A. H. 'Stochastic processes and filtering theory', Academic Press, 1973.

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

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

[4] Masreliez, C. J. 'Approximate non-Gaussian filtering with linear state and observation relations', *IEEE Trans. Auto. Control*, 1975, **20**, pp. 107–110.

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

[6] Bucy, R. S. 'Bayes theorem and digital realiasation for nonlinear filters.', *Journal of Austronautical Science.*, 1969, **17**, pp. 80–94.

[7] Kramer, S. C. and Sorenson, H. W. 'Recursive Bayesian estimation using piece-wise constant approximations', *Automatica*, 1988, **24**, pp. 789–801.

[8] Sorenson, H. W. 'Recursive estimation for nonlinear dynamic systems', *in* Spall, J. C., editor, 'Bayesian analysis of time series and dynamic models', Dekker, 1988.

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

[10] Pole, A. and West, M. 'Efficient Bayesian learning in dynamic models', *Journal of Forecasting*, 1990, **9**, pp. 119–136.

[11] Carlin, B. P., Polson, N. G., and Stoffer, D. S. 'A Monte Carlo approach to nonnormal and nonlinear state-space modelling', *JASA*, 1992, **87**, pp. 493–500.

[12] West, M. 'Modelling with mixtures', *Bayesian Statistics*, 1992, **4**, pp. 503–524.

[13] Mueller, P. 'Posterior Integration in Dynamic models', Technical Report 92-A13, Institute of Statistics and Decision Sciences, Duke University, Durham, NC 27706, 1992.

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

[15] Isard, M. and Blake, A. 'Contour tracking by stochastic propagation of conditional density', *in* 'Proc. European Conf. Computer Vision', pages 343–356, Cambridge UK, 1996.

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

[17] BERZUINI, C., BEST, N. G., GILKS, W. R., AND LARIZZA, C. ‘Dynamic conditional independence models and Markov chain Monte Carlo methods.’, *J. Am. Statist. Assoc. To appear*, 1998.

[18] LIU, J. S. AND CHEN, R. ‘Monte Carlo methods for dynamic systems’, *J. Am. Statist. Assoc. To appear*, 1998.

[19] DOUCET, A. ‘On sequential simulation-based methods for Bayesian filtering’, Technical report, University of Cambridge, Cambridge, CB2 1PZ, Department of Engineering, 1998.

[20] RUBIN, D. ‘Comment on ‘The calculation of posterior distributions by data augmentation’ by Tanner, M. A. and Wong W. H.’, *J. Am. Statist. Assoc.*, 1987, **82**, pp. 543.

[21] GORDON, N., SALMOND, D., AND EWING, C. ‘Bayesian state estimation for tracking and guidance using the bootstrap filter’, *Journal of Guidance, Control and Dynamics*, November 1995, **18**, pp. 1434–1443.

[22] PITT, M. K. AND SHEPHARD, N. ‘Filtering via simulation: auxiliary particle filters’, Technical report, Nuffield College, Oxford University, Oxford OX1 1NF, UK, September 1997.

[23] FISHMAN, G. S. ‘Monte Carlo—Concepts, Algorithms and Applications.’, New York: Springer, 1996.

[24] SMITH, A. F. M. AND GELFAND, A. E. ‘Bayesian Statistics Without Tears: A Sampling-Resampling Perspective’, *The American Statistician*, 1992, **46**, pp. 84–88.

[25] HAMMERSLEY, J. M. AND HANDSCOMB, D. C. ‘Monte Carlo Methods.’, London: Methuen, 1964.

[26] COCHRAN, W. G. ‘Sampling techniques.’, London: Wiley, 1963.

[27] CRISAN, D. AND LYONS, T. ‘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, 1997.

[28] GERONTIDES, I. AND SMITH, R. L. ‘Monte Carlo generation of order statistics from a general distribution’, *Applied Statistics*, 1982, **31**, pp. 236–243.

[29] GILKS, W. R., RICHARDSON, S., AND SPIEGELHALTER, D. J. ‘Markov Chain Monte Carlo in practice’, London: Chapman and Hall, 1996.

[30] CARPENTER, J. R., CLIFFORD, P., AND FEARNHEAD, P. ‘Sampling Strategies for Monte Carlo filters of non-linear systems’, *in* ATHERTON, D. P., editor, ‘IEE Colloquium Digest, Target Tracking and Control, DERA, Malvern’, volume 6, 253, pages 1–3, London, 1996.

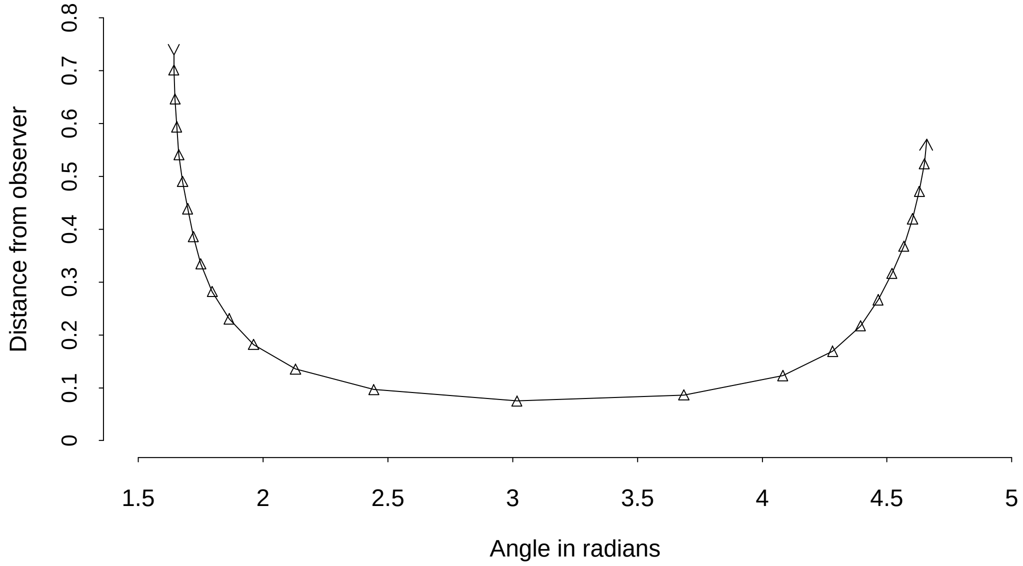

FIGURE 1

**Caption for Figure 1:**

Trajectory used to compare particle filters in the Section 6. The object passes from left to right. There are 24 time steps. It is when it passes close to the origin, at step 10, that the problem is most non-linear and the particle filters start to degenerate.

| Step | Improved Reweighted | | Multinomial Reweighted | | Two-stage sampling | | Standard SIR | |
|---|---|---|---|---|---|---|---|---|
| | Radius | Angle | Radius | Angle | Radius | Angle | Radius | Angle |
| 2 | 4545 | 4869 | 4545 | 4869 | 4545 | 4869 | 4545 | 4869 |
| 4 | 1190 | 2614 | 794 | 1648 | 588 | 1317 | 714 | 1192 |
| 6 | 236 | 866 | 147 | 602 | 90 | 348 | 132 | 355 |
| 8 | 106 | 815 | 85 | 563 | 48 | 339 | 47 | 263 |
| 10 | 81 | 521 | 62 | 399 | 31 | 212 | 31 | 179 |
| 12 | 25 | 684 | 20 | 538 | 11 | 249 | 9 | 162 |
| 14 | 14 | 2757 | 11 | 2387 | 6 | 1540 | 5 | 479 |
| 16 | 18 | 1377 | 14 | 1051 | 7 | 668 | 5 | 267 |
| 18 | 17 | 4538 | 14 | 3701 | 7 | 2871 | 5 | 1803 |
| 20 | 17 | 775 | 14 | 640 | 7 | 350 | 5 | 243 |
| 22 | 19 | 436 | 15 | 360 | 7 | 204 | 4 | 109 |
| 24 | 19 | 2394 | 15 | 1560 | 7 | 967 | 5 | 621 |

TABLE 1

**Caption for Table 1**

Effective sample sizes (rounded to the nearest integer) obtained using various particle filters (see text). The mean angle or bearing is well estimated but in all cases there is severe loss of information about the range from step 10 onwards.