# Improvement Strategies for Monte Carlo Particle Filters

*Simon Godsill and Tim Clapp*
Signal Processing Group, University of Cambridge

## 1 Introduction

The particle filtering field has seen an upsurge in interest over recent years and accompanying this a number of enhancements to the basic techniques have been suggested in the literature. In this paper we collect together a group of these developments which seem to be particularly important for time series applications and give a broad discussion of the methods, showing the interrelationships between them. We firstly present a general importance sampling framework for the filtering/smoothing problem and show how the standard techniques can be obtained from this general approach. In particular, we show that the auxiliary particle filtering methods of (Pitt and Shephard: this volume) fall into the same general class of algorithms as the standard bootstrap filter of (Gordon, Salmond and Smith 1993). We then develop the ideas further and describe the role of MCMC resampling, as proposed by (Gilks and Berzuini: this volume) and (MacEachern, Clyde and Liu 1999). Finally, we present a generalisation of our own in which MCMC resampling ideas are used to traverse a sequence of 'bridging' densities which lie between the prediction density and the filtering density. In this way it is hoped to reduce the variability of the importance weights by attempting a series of smaller, more manageable moves at each time step.

Much of the particle filtering literature has been concerned with filtering for non-linear models in tracking applications. There are many other important contexts where sequential methods are needed, and here we explore a class of time-evolving models which can be used to explore the spectral evolution of a wide range of datasets: the time-varying autoregression. Autoregressions have been used to model a wide range of phenomena from earthquake data and eeg traces to financial time series. We have applied them extensively to the analysis and processing of speech and audio data (Godsill and Rayner 1998a)(Godsill and Rayner 1998b)(Godsill 1997), where it is typically important to model an additive background noise component in addition to the desired signal itself. Moreover, in such applications it is valuable to track the time-varying spectral structure which is present in the data, and this can be achieved by extending the standard models to have time-varying parameters, see (Rajan, Rayner and Godsill 1997)(Kitagawa and Gersch 1996)(West and Harrison 1997)(West, Prado and Krystal 1999)(Prado, West and Krystal 1999) for examples of their theory and application in a statistical context. In this paper we study a simple motivating TVAR model which captures the basic characteristics of many time-varying datasets. A Gaussian random walk is used to model the autoregressive parameter variation with time and signals are assumed observed in white Gaussian noise. More elaborate models which incorporate stability constraints and more physically-based time evolution structures are discussed in (Godsill, Doucet and West 2000). Here we work with the most simple model since it is straightforward to construct efficient implementations of all of the above variants on particle filtering and we describe in detail the resulting algorithms. Simulations are then performed in order to compare the merits of the various approaches, both in terms of estimation performance and computational burden. In our initial discussions we will assume the standard Markovian state-space model

$$\begin{aligned} x_t &\sim f(x_t|x_{t-1}) && \text{State evolution density} \\ y_t &\sim g(y_t|x_t) && \text{Observation density} \end{aligned}$$

where $\{x_t\}$ are unobserved states of the system and $\{y_t\}$ are observations made over some time interval $t \in \{1, 2, ..., T\}$. $f(.|.)$ and $g(.|.)$ are prespecified state evolution and observation densities. The initial state has density $p(x_1)$.

## 2 General Sequential Importance Sampling

Here we formulate the sequential importance sampling method in terms of updates to the smoothing density. Once we have samples drawn from the smoothing density it is straightforward to discard those which are not required if filtering is the main objective. Suppose we have at time $t$ weighted particles $\{x_{1:t}^{(i)}, w_t^{(i)}; i = 1, 2, ..., N\}$ drawn from the smoothing density $p(x_{1:t}|y_{1:t})$. We can represent this information using the 'particulate' approximation to the density:

$$p(x_{1:t}|y_{1:t}) \simeq \sum_{i=1}^{N} w_t^{(i)} \delta(x_{1:t} - x_{1:t}^{(i)}),$$

where $\sum_{i=1}^{N} w_t^{(i)} = 1$. In order to update the smoothing density from time $t-1$ to time $t$, we factorise it as follows:

$$p(x_{1:t}|y_{1:t}) = p(x_{1:t-1}|y_{1:t-1}) \times \frac{g(y_t|x_t) f(x_t|x_{t-1})}{p(y_t|y_{1:t-1})} \tag{2.1}$$

where the denominator is simply a fixed normalising constant. The derivation presented here is slightly non-standard, involving an importance function over all of the smoothed states. This allows us to consider a range of methods, including those with elements of smoothing as well as filtering, within one single framework. For more standard interpretations, see the review chapters of this book.

Ideally we wish to update the particles at time $t-1$ in order to obtain joint draws from $p(x_{1:t}|y_{1:t})$. Using importance sampling we can achieve this by generating states from an importance density $q(x_{1:t})$ which may implicitly depend upon the particles at time $t-1$ and observed data $y_{1:t}$, as required. The importance weight, from equation (2.1), is then given by:

$$w_t \propto p(x_{1:t-1}|y_{1:t-1}) \times \frac{g(y_t|x_t)f(x_t|x_{t-1})}{q(x_{1:t})}$$

For standard filtering problems it is usual to fix the values of the particles at time $t-1$ and generate the new states $x_t$ conditional upon those particles. The importance function for each particle can in this case take the form $q(x_{1:t}) = q(x_t|x_{1:t-1}) \sum_{i=1}^{N} v_{t-1}^{(i)} \delta(x_{1:t-1} - x_{1:t-1}^{(i)})$, that is we draw $x_{1:t-1}$ from a particulate representation with normalised weights $v_{t-1}^{(i)}$ and then generate $x_t$ conditionally from a proposal $q(x_t|x_{1:t-1})$. Substituting this and the particulate approximation for $p(x_{1:t-1}|y_{1:t-1})$ into the weight expression leads to:

$$w_t \propto \frac{w_{t-1}}{v_{t-1}} \times \frac{g(y_t|x_t)f(x_t|x_{t-1})}{q(x_t|x_{1:t-1})}$$

where $x_{1:t} = \{x_{1:t-1}, x_t\}$ is a particle drawn from the joint distribution $q(x_{1:t})$. The weights $v_{t-1}$ serve to amplify or diminish the numbers of different particles in the population, thus allowing the importance sampler the potential to neglect particles with low probability and increase the numbers of particles with high probability. A standard approach chooses $v_t = w_t$, in which case particles are selected with probability proportional to their weights in the approximation to $p(x_{1:t-1}|y_{1:t-1})$. This case is equivalent to the bootstrap filter of (Gordon et al. 1993) in which particles $x_{1:t-1}$ are resampled according to their weights $w_{t-1}$, prior to making a conditional draw for the new state $x_t \sim f(x_t|x_{t-1})$. However, particles which had large weights at time $t-1$ should not necessarily have large weights when propagated to $t$, in the light of an 'informative' observation $y_t$. An improved choice of weighting function may then be $v_{t-1} \propto p(x_{1:t-1}|y_{1:t})$, since then particles are amplified or diminished directly according to their posterior distribution conditional upon all the data, including the new point $y_t$. This is the principle behind the auxiliary particle filter (Pitt and Shephard: this volume), although in practice $p(x_{1:t-1}|y_{1:t})$ will be unavailable analytically and so $v_{t-1}$ must be approximated in their scheme (see later for implementational details of the TVAR model).

**A note on selection schemes.** The general framework described above involves a random resampling step, in which particles are resampled randomly with replacement with probabilities $\{v_t^{(i)}\}$. This random resampling introduces unnecessary additional Monte Carlo variation and can be replaced with any partly deterministic scheme such that the expected number selected of each particle equals $Nv_t$. Such schemes can be designed to be much more computationally efficient than the fully random selection scheme which simply selects particles independently from the multinomial distribution defined by the weights. There is a number of such schemes, see (Kitagawa 1996)(Carpenter, Clifford and Fearnhead 1999)(Liu and Chen 1998) and we do not discuss the merits of each here. Simply note that in all of our simulations the residual resampling method of (Liu and Chen 1998) is used. A further option is to not perform selection at every time step, rather accumulating weights through time and only performing selection when some appropriate measure of weight degeneracy, such as effective sample size (Liu and Chen 1995), reaches some threshold. In all of our simulations a selection step is carried out at every iteration, although we note that further improvements could be made by less frequent selection operations.

In any of the above variants on the sequential importance sampling /resampling schemes we have at time $t$ an approximate set of draws from the joint smoothing density, $p(x_{1:t}|y_{1:t})$. In practice, for all of the schemes, excepting perhaps the most general version which allows an importance function $q(x_{1:t})$ over the complete state sequence, we expect a high degree of degeneracy between the smoothed state sequences at times significantly less than $t$. This is because the importance function leaves the state sequences up to time $t-1$ unchanged in the move to $t$. This can lead to problems in pure filtering problems (where only $p(x_t|y_{1:t})$ is of interest) which become even more severe in cases where an element of smoothing is also required. Another problem where this degeneracy is particularly troublesome is the case of models with fixed (non-dynamic) hyperparameters. Here the smoothing density is augmented by fixed dimension parameters $\theta$ to $p(x_{1:t}, \theta|y_{1:t})$, which can be factorised as:

$$p(x_{1:t}, \theta|y_{1:t}) = p(x_{1:t-1}, \theta|y_{1:t-1}) \times \frac{g(y_t|x_t, \theta) f(x_t|x_{t-1}, \theta)}{p(y_t|y_{1:t-1})}$$

All of the above ideas can be applied in principle to this case, augmenting the particles to include $\theta$ as well as $x_{1:t}$ at each time step: $\{x_{1:t}^{(i)}, \theta^{(i)}, w_t^{(i)}; i = 1, 2, ..., N\}$. The degeneracy problem becomes particularly severe here, however, since there is no means of generating new $\theta$ values as time proceeds when the particulate approximation is employed. We do not address this fixed parameter issue in our simulations, rather noting that it is an important problem which can be tackled using the Markov chain methods of the next sections or with kernel density ideas (Liu and West: this volume).

In the next section we describe techniques which allow for regeneration of new state values through the use of Markov chains and hence go some way towards solving the degeneracy issues associated with standard particle filtering methods.

# 3 Markov chain moves

Using the methods of the previous section we have the means to generate approximate draws from $p(x_{1:t}|y_{1:t})$. As stated, however, there can be problems of degeneracy in the particle filter when the target distribution $p(x_{1:t}|y_{1:t})$ does not overlap significantly with the prediction distribution $p(x_{1:t}|y_{1:t-1})$. One way to improve this situation is to incorporate Markov chain moves into the scheme (MacEachern et al. 1999)(Gilks and Berzuini: this volume). Since we wish particles to be drawn from $p(x_{1:t}|y_{1:t})$, it would seem reasonable to design Markov chain transition kernels $T(x_{1:t}, dx'_{1:t})$, having $p(x_{1:t}|y_{1:t})$ as stationary distribution. There are many well known ways for achieving this (Gilks, Richardson and Spiegelhalter 1996), including the Gibbs sampler and Metropolis-Hastings methods. We will explicitly develop MCMC samplers for the TVAR model in the simulation section. Intuitively, if the particles $\{x_{1:t}^{(i)}\}$ are truly drawn from $p(x_{1:t}|y_{1:t})$, then the Markov chain kernel applied to any of the particles will simply generate new state sequences which are also drawn from the desired distribution. Moreover, degeneracy will be less of a problem since particles can move independently from one another to different points in the state space. More importantly, perhaps, if $\{x_{1:t}^{(i)}\}$ are not drawn accurately from $p(x_{1:t}|y_{1:t})$, owing to the approximation of importance sampling with finite $N$, the Markov chain will nudge the particles closer to the target distribution. Thus we have a means of improving both degeneracy amongst particles and the accuracy of the samples in the particle cloud. In practice for filtering problems with large datasets the transition kernels will not operate on the whole of the state history, but rather on recent elements back to some fixed or varying time lag $l$.

These methods also help significantly with the fixed parameter problem, since a Markov chain transition kernel can be applied for generating new values of the non-dynamic parameter $\theta$. Typically $\theta$ might be drawn from its conditional distribution $p(\theta|x_{1:t}, y_{1:t})$ either exactly in a single move step if the distributions allow (Clapp and Godsill 1999), or iteratively using a Markov chain kernel. One troublesome aspect of this, however, is that the conditional distribution will in general depend upon the entire state history $x_{1:t}$, which means that computational burden and memory requirements will be a problem for large datasets. For certain cases, however, such as the conditionally Gaussian parameter model, this distribution may be summarised by simple sufficient statistics which are then efficiently updated at each time step.

### 3.1 The Use of Bridging Densities with MCMC moves

The combination of importance sampling with Markov chain moves is powerful, and helps to reduce sigificantly the problems associated with sequential Monte Carlo. However, there are situations where a very large number of MCMC iterations would be required in order to reach the target distribution. In particular this can occur when the likelihood for the new data point is centred far from the points sampled from the importance distribution $q(x_t|x_{1:t-1})$. The MCMC sampler may then have to converge from a set of atypical points lying in the tails of the target distribution.This will be an expensive procedure, since it must be repeated for all samples, and there may be no reliable way of judging whether convergence has actually been achieved.

Here we explore a new version of the sequential Monte Carlo schemes outlined above in which a sequence of bridging densities is placed between the initial distribution at time $t-1$ and the target distribution at time $t$. It is hoped that by replacing a single large transition with a series of smaller transitions the Monte Carlo variation of the importance weights can be reduced. All of the above ideas combining importance sampling with Markov chain moves can be used to good effect within the new scheme. The initial distribution in the sequence is denoted by $\pi_0 = q(x_{1:t})$ and the final distribution $\pi_{M+1} = p(x_{1:t}|y_{1:t})$. Intervening bridging distributions $\pi_1, \pi_2, ..., \pi_M$ are then defined according to some appropriate schedule such as:

$$\pi_m(x_{1:t}) \propto q(x_{1:t})^{\alpha_m} p(x_{1:t}|y_{1:t})^{1-\alpha_m} \tag{3.1}$$

with $1 > \alpha_1 > \alpha_2 ... > \alpha_M > 0$. For example, if the importance distribution is of the form $q(x_{1:t}) = f(x_t|x_{t-1}) \sum_{i=1}^N w_{t-1}^{(i)} \delta(x_{1:t-1} - x_{1:t-1}^{(i)})$, which corresponds to the distribution used in the bootstrap filter of (Gordon et al. 1993), then we have

$$\pi_m(x_{1:t}) \propto p(x_{1:t-1}|y_{1:t-1}) f(x_t|x_{t-1}) g(y_t|x_t)^{1-\alpha_m} \tag{3.2}$$

and it is evident that the bridging densities gradually introduce the likelihood as $g(y_t|x_t)^{1-\alpha_m}$, starting with a flat likelihood ($\alpha_0 = 1$) and gradually 'sharpening' it until $\alpha_{M+1} = 0$.

The task is now to move the particle cloud through this sequence of densities by whatever means are available. A MCMC scheme for achieving this in a batch context for multimodal target distributions, called annealed importance sampling, has been introduced by (Neal 1998). In our sequential context an algorithm can be devised as follows. Firstly, generate states from the proposal distribution $q(x_{1:t})$ and denote these $\{x_{1:t}^{(i)(0)}, w_t^{(i)(0)}\}$ with $w_t^{(i)(0)} = w_{t-1}^{(i)}$. Then at step $m$, for $m = 1, ..., M+1$, compute the incremental importance weight for the transition between $\pi_{m-1}(x_{1:t})$ and $\pi_m(x_{1:t})$, i.e.

$$w_t^{(i)(m)} \propto w_t^{(i)(m-1)} \frac{\pi_m(x_{1:t}^{(i)(m-1)})}{\pi_{m-1}(x_{1:t}^{(i)(m-1)})}$$

We now have nominally a weighted sample from $\pi_m(x_{1:t})$. We can optionally perform a selection step here with probabilities $\{ w_{t+1}^{(i)(m)}\}$ if the weights are too degenerate and reset $\{ w_{t+1}^{(i)(m)}\}$ to uniform. Now, exactly as in the MCMC schemes above for a single target density, we can improve the particle stock by drawing $x_{1:t}^{(i)(m)}$ from $T_m(x_{1:t}^{(i)(m-1)}, dx'_{1:t})$, where $T_m(x_{1:t}^{(i)(m-1)}, dx'_{1:t})$ is a Markov chain transition kernel designed to have $\pi_m(x_{1:t})$ as invariant distribution. The importance weights are hence unchanged. Finally, after step $M+1$, we have a weighted sample from the target distribution $\pi_{M+1}(x_{1:t}) = p(x_{1:t}|y_{1:t})$, as required, with weights $w_t^{(i)} = w_t^{(i)(M+1)}$. (Neal 1998) proves that such a procedure generates valid importance weights for the final samples $\{x_{1:t}^{(i)(M+1)}\}$ according to the target distribution $p(x_{1:t}|y_{1:t})$.

Note that the special case with $M = 0$ (i.e. no bridging densities) corresponds to the MCMC resampling algorithms of (MacEachern et al. 1999) (Gilks and Berzuini: this volume). We have generalised Neal's procedure slightly here by allowing an optional resampling at each intermediate step. This has sometimes been found to be beneficial.

# 4 Simulation example: TVAR Model in Noise

The time-varying autoregressive (TVAR) model can be used to model parametrically a signal with time-varying frequency content. Here we work with the simplest TVAR structure and develop particle filtering approaches based upon the improvement strategies described above. It is assumed that the TVAR signal is observed in additive white Gaussian noise, which models any background noise present in the measurement environment. This aspect of the model is important in acoustical applications such as speech and audio (Godsill and Rayner 1998a)(Godsill and Rayner 1998b)(Godsill 1997) and adds an interesting challenge to the parameter estimation process. The TVAR signal process $\{z_t\}$ is generated as a linear weighted sum of previous signal values:

$$z_t = \sum_{i=1}^{P} a_{i,t} z_{t-i} + e_t = a_t^T z_{t-1:t-P} + e_t \tag{4.1}$$

Here $a_t = (a_{1,t}, a_{2,t}, ..., a_{P,t})^T$ is the $P^{\text{th}}$ order AR coefficient vector at time $t$. The innovation sequence $\{e_t\}$ is assumed independent and Gaussian with time-varying variance $\sigma_{e_t}^2$. Hence we may write the conditional density for $z_t$ as:

$$f(z_t|z_{t-1:t-P}, a_t, \sigma_{e_t}^2) = \mathcal{N}\left(a_t^T z_{t-1:t-P}, \sigma_{e_t}^2\right)$$

The signal is assumed to be observed in independent Gaussian noise:

$$y_t = z_t + v_t \tag{4.2}$$

so that we may write the density for the observation $y_t$ as $g(y_t|z_t, \sigma_{v_t}) = \mathcal{N}\left(z_t, \sigma_{v_t}^2\right)$, where once again $\sigma_{v_t}^2$ may be time-varying. For our simulations, a Gaussian random walk model is assumed for the log-standard deviations $\phi_{e_t} = \log(\sigma_{e_t})$ and $\phi_{v_t} = \log(\sigma_{v_t})$:

$$f(\phi_{e_t}|\phi_{e_{t-1}}, \sigma_{\phi_e}^2) = \mathcal{N}\left(\phi_{e_{t-1}}, \sigma_{\phi_e}^2\right), \tag{4.3}$$
$$f(\phi_{v_t}|\phi_{v_{t-1}}, \sigma_{\phi_v}^2) = \mathcal{N}\left(\phi_{v_{t-1}}, \sigma_{\phi_v}^2\right) \tag{4.4}$$

The model now requires specification of the time variation in $a_t$ itself. One of the simplest choices of all is a first order autoregression directly on the coefficients

$$f(a_t|a_{t-1}, \sigma_a^2) = \mathcal{N}\left(\alpha a_{t-1}, \sigma_a^2 I_P\right) \tag{4.5}$$

where $\alpha$ is a constant just less than 1.

More elaborate schemes of this sort are possible, such as a smoothed random walk involving AR coefficients from further in the past, or a non-diagonal covariance matrix, but none of these modifications make a significant difference to the computations required for the particle filter. In this paper we do not consider the stability of the TVAR model, see (Godsill et al. 2000) for more detailed consideration of this issue.

The state space TVAR model is now fully specified. The unobserved state vector is $x_t = (z_{t:t-P+1}, a_t, \phi_{e_t}, \phi_{v_t})$. Hyperparameters $\sigma_a^2$, $\sigma_{\phi_e}^2$ and $\sigma_{\phi_v}^2$ are assumed pre-specified and fixed in all the simulations. The initial state probability is specified as iid Gaussian. Note that we will refer to $\phi_{e_t}$ and $\phi_{v_t}$ interchangeably with $\sigma_{e_t}$ and $\sigma_{v_t}$ in the subsequent discussions, as is convenient from the context. Also, we will refer to $x_t$ or its individual components $(z_{t:t-P+1}, a_t, \phi_{e_t}, \phi_{v_t})$ interchangeably.

## 4.1 Particle Filter Algorithms for TVAR Models

In the models specified above there is some rich analytic structure. For example, conditionally on the signal sequence and noise variance sequences the AR parameter sequence $\{a_t\}$ is Gaussian and can be estimated using the Kalman filter recursions. Similarly, the signal sequence $\{z_t\}$ is Gaussian conditionally upon $\{a_t\}$ and the variance sequences and can also be estimated using the Kalman filter. This suggests a range of 'Rao-Blackwellised' versions of the particle filter in which one or other of the sequences $\{a_t\}$ or $\{z_t\}$ is marginalised sequentially from the problem using the Kalman filter. These approaches lead to lower-dimensional state spaces for the particle filter and can be expected to lead to greater robustness (Doucet, Godsill and Andrieu 1999). However, they come with a computational penalty, since Kalman filter recursions must be applied at every time step for every particle. We do not explore these approaches here, both for reasons of computational complexity and because we lose some of the generality of the particle filter: a small change to the model structure by introducing some non-linearity or non-Gaussianity (such as those in (Godsill et al. 2000)) would immediately destroy the analytic structure present and invalidate the approach. Some of these Rao-Blackwellised versions of the particle filter are investigated for a speech application in (Vermaak, Andrieu, Doucet and Godsill 1999).

Firstly, consider the choice of importance function for the various filtering approaches. The 'prior' importance function, as used in the original bootstrap filter implementations (Gordon et al. 1993) requires a random draw from the state transition density, as follows:

$$\phi_{e_t} \sim \mathcal{N}\left(\phi_{e_{t-1}}, \sigma^2_{\phi_e}\right), \quad \phi_{v_t} \sim \mathcal{N}\left(\phi_{v_{t-1}}, \sigma^2_{\phi_v}\right), \quad a_t \sim \mathcal{N}\left(\alpha a_{t-1}, \sigma^2_a I_P\right) \tag{4.6}$$

$$z_t \sim \mathcal{N}\left(a_t^T z_{t-1:t-P}, \sigma^2_{e_t}\right) \tag{4.7}$$

where $I_P$ denotes the $P$-dimensional identity matrix.

This is a very simple and efficient procedure and in the basic bootstrap filtering framework an importance weight can be computed as:

$$w_t \propto g(y_t|z_t, \sigma_{v_t}) \tag{4.8}$$

In fact, we have found algorithms based on this procedure to perform very poorly, largely as a result of step (4.6), which can generate 'unstable' realisations of the AR parameters (a sufficient condition for stability of a TVAR model is that the instantaneous poles, i.e. the roots of the polynomial $(1 - \sum_{i=1}^{P} a_{i,t} p^{-i})$, lie strictly within the unit circle – when poles move outside the unit circle, 'unstable' oscillatory behaviour is observed in the generated signal). Hence, when the final step (4.7) is performed, large and oscillating values of the signal are generated and the filter can diverge entirely from the correct state sequence. This problem can be fixed by using a slightly cleverer importance function which takes account of the fact that $z_t$ is conditionally Gaussian. We can then sample $z_t$ from its conditional posterior distribution $p(z_t|x_{t-1}, y_t, a_t, \sigma_{e_t}, \sigma_{v_t}) \propto f(z_t|z_{t-1:t-P}, a_t, \sigma_{e_t}) g(y_t|z_t, a_t, \sigma_{v_t})$, which is Gaussian. The incorporation of the effect of $y_t$ in this manner is generally found to be sufficient to stop the filter from diverging due to stability problems. Working through the algebra, step (4.7) is replaced by

$$z_t \sim \mathcal{N}\left(\widehat{z}_t, 1/\phi_t\right) \tag{4.9}$$

where

$$\widehat{z}_t = \theta_t/\phi_t, \qquad \theta_t = y_t/\sigma^2_{v_t} + a_t^T z_{t-1:t-P}/\sigma^2_{e_t} \quad \text{and} \quad \phi_t = 1/\sigma^2_{v_t} + 1/\sigma^2_{e_t} \tag{4.10}$$

The resulting importance function is then summarised as:

$$q(x_t|x_{t-1}, y_t) = f(\phi_{v_t}|\phi_{v_{t-1}}, \sigma^2_{\phi_v}) \times f(\phi_{e_t}|\phi_{e_{t-1}}, \sigma^2_{\phi_e}) f(a_t|a_{t-1}, \sigma^2_a) p(z_t|x_{t-1}, y_t, a_t, \sigma_{e_t}, \sigma_{v_t}) \tag{4.11}$$

and the importance weight is modified accordingly to:

$$w_t \propto \frac{g(y_t|x_{t-1})f(x_t|x_{t-1})}{q(x_t|x_{t-1},y_t)} \tag{4.12}$$
$$\propto p(y_t|z_{t-1:t-P},a_t,\sigma_{e_t}^2,\sigma_{v_t}^2) \tag{4.13}$$
$$\propto \frac{(2\pi)^{1/2}}{\phi_t^{1/2}} N(y_t|\widehat{z}_t,\sigma_{v_t}^2) N\left(\widehat{z}_t|a_t^T z_{t-1:t-P},\sigma_{e_t}^2\right) \tag{4.14}$$

where $N(a|\mu,\sigma^2)$ denotes the Gaussian density function with mean $\mu$ and variance $\sigma^2$, evaluated at $a$. This form of importance function involves only a modest computational overhead and is adopted in all of the ensuing simulations. It is somewhat more restrictive than the general form since it requires us to be able to marginalise $z_t$, but we note that this will be possible for any non-Gaussian noise distribution which can be reparameterised as a scale mixture of Gaussians, see (Godsill and Rayner 1998a)(Godsill and Rayner 1998b)(Godsill 1997) (Shephard 1994) for discussion of such reparameterisations in related modelling contexts. This form of importance function (4.11) forms the basic building block in all of our algorithms.

## 4.2 Bootstrap (SIR) Filter

The bootstrap filter (Gordon et al. 1993) is constructed using sampling importance resampling (SIR) with the importance function defined in (4.11). The algorithm is summarised as[1]:

1. Set $t = 0$. Draw $x_0^{(i)} \sim p(x_0)$, $i = 1, ..., N$
2. For $t = 1$ to $T$
   - $x_t^{(i)} \sim q(x_t|x_{t-1}^{(i)}, y_t)$, $i = 1, ..., N$, see (4.11)
   - $w_t^{(i)} \propto p(y_t|z_{t-1:t-P}^{(i)}, a_t^{(i)}, \sigma_{e_t}^{2(i)}, \sigma_{v_t}^{2(i)})$, $i = 1, ..., N$, $\sum_{i=1}^{N} w_t^{(i)} = 1$, see (4.14)
   - Select with replacement from $\{x_t^{(i)}\}$ with probability $\{w_t^{(i)}\}$
3. End

[1]Note that in the implementation the number of particles is fixed at $N$, whereas the full scheme of (Gordon et al. 1993) allows for prior boosting, with a larger number of particles generated at the Sample stage

## 4.3 Auxiliary Particle Filter (APF)

As described in section 2, the auxiliary particle filter (Pitt and Shephard: this volume) attempts to reduce the variability of the importance weights by resampling the particles at time $t-1$ with probability close to $p(x_{t-1}|y_{1:t})$, i.e. allowing one step of lookahead to improve the particle stock at time $t$. In a basic SIR implementation of the APF, $p(x_{t-1}|y_{1:t})$ is approximated by making a prediction $\widehat{x}_t$ of $x_t$ from each particle $x_{t-1}$, which is then substituted in as a single point mass approximation to the transition density $f(x_t|x_{t-1})$. The first stage of the process then selects $x_{t-1}$ with replacement according to the approximation of $p(x_{t-1}|y_{1:t})$. The second stage proceeds as in the basic bootstrap filter, generating new states $x_t$ randomly from the state transition density $f(x_t|x_{t-1})$. The importance weights are straightforwardly computed (Pitt and Shephard: this volume) as $w_t \propto g(y_t|x_t)/g(y_t|\widehat{x}_t)$ . In our implementation, we generate the new states as before from (4.11) in order to introduce greater robustness. The weights are then modified to:

$$w_t = \frac{p(y_t|z_{t-1:t-P}, a_t, \sigma^2_{e_t}, \sigma^2_{v_t})}{p(y_t|z_{t-1:t-P}, \widehat{a}_t, \widehat{\sigma}^2_{e_t}, \widehat{\sigma}^2_{v_t})} \tag{4.15}$$

where $\widehat{a}_t, \widehat{\sigma}^2_{e_t}, \widehat{\sigma}^2_{v_t}$ are deterministic predictions of $a_t, \sigma^2_{e_t}, \sigma^2_{v_t}$ made from the previous states. We use here simply the mode of the state transition densities, i.e. $\widehat{a}_t = a_{t-1}$, $\widehat{\sigma}^2_{e_t} = \sigma^2_{e_{t-1}}$, $\widehat{\sigma}^2_{v_t} = \sigma^2_{v_{t-1}}$, which correponds to our expectation that AR coefficients and noise variances will change slowly with time. Notice that for this importance function the generated values of $z_t$ once again do not appear in the weight equation: we have eliminated a possible source of extra Monte Carlo variation. The APF implementation can be summarised as:

1. Set $t = 0$. Draw $x_0^{(i)} \sim p(x_0)$, set $w_0^{(i)} = 1/N, \ \ i = 1, ..., N$
2. For $t = 1$ to $T$
   - $\widehat{a}_t^{(i)} = \alpha a_{t-1}^{(i)}$, $\widehat{\sigma}_{e_t}^{2(i)} = \sigma_{e_{t-1}}^{2(i)}$, $\widehat{\sigma}_{v_t}^{2(i)} = \sigma_{v_{t-1}}^{2(i)}, \ \ i = 1, ..., N$
   - $w_t^{(i)} \propto w_{t-1}^{(i)} p(y_t|z_{t-1:t-P}^{(i)}, \widehat{a}_t^{(i)}, \widehat{\sigma}_{e_t}^{2(i)}, \widehat{\sigma}_{v_t}^{2(i)}), \ \ \sum_{i=1}^{N} w_t^{(i)} = 1, \ \ i = 1, ..., N$
   - Select with replacement from $\{x_{t-1}^{(i)}\}$ with probability $\{w_t^{(i)}\}$
   - $x_t^{(i)} \sim q(x_t|x_{t-1}^{(i)}, y_t)$, $i = 1, ..., N$, see (4.11)
   - $w_t^{(i)} \propto \frac{p(y_t|z_{t-1:t-P}^{(i)}, a_t^{(i)}, \sigma_{e_t}^{2(i)}, \sigma_{v_t}^{2(i)})}{p(y_t|z_{t-1:t-P}^{(i)}, \widehat{a}_t^{(i)}, \widehat{\sigma}_{e_t}^{2(i)}, \widehat{\sigma}_{v_t}^{2(i)})}, \ \ \sum_{i=1}^{N} w_t^{(i)} = 1$, see (4.14)
3. End

We can see that there is a modest computational overhead in the weight computations compared with the standard bootstrap filter, but for some problems this may be offset by gains in estimation performance.

## 4.4 MCMC Resampling

In order to implement any of the MCMC based schemes discussed above we need to design an effective Markov chain transition kernel with $p(x_t|x_{t-1}, y_t)$ as its invariant distribution. For the model chosen here it is straightforward to implement a Gibbs sampler which draws random samples from the full conditionals for $z_t$, $a_t$, $\sigma^2_{e_t}$ and $\sigma^2_{v_t}$ in turn. Similar samplers have been found to work well in the batch-based case (Godsill and Rayner 1998a)(Godsill and Rayner 1998b) (Godsill 1997) and can be expected to move very rapidly towards the target distribution when implemented for a single state $x_t$. We already have the full conditional for $z_t$ in (4.9). The full conditional for $a_t$ is also Gaussian:

$$p(a_t|z_t, \sigma^2_{e_t}, \sigma^2_{v_t}, x_{t-1}, y_t) = \mathcal{N}\left(\Phi_t^{-1}\Theta_t, \Phi_t^{-1}\right) \tag{4.16}$$

$$\begin{aligned} \Phi_t &= I_P/\sigma_a^2 + Z_{t-1}/\sigma^2_{e_t} && (4.17)\\ \Theta_t &= \alpha a_{t-1}/\sigma_a^2 + z_{t-1:t-P} z_{t-1}/\sigma^2_{e_t} && (4.18)\\ Z_{t-1} &= z_{t-1:t-P} z^T_{t-1:t-P} && (4.19) \end{aligned}$$

Hence it is straightforward to implement the Gibbs sampler for these two components using standard Gaussian sampling techniques. Note however that sampling for $a_t$ is computationally far more demanding than any of the other steps we have so far considered when the model order $P$ is high. The full conditionals for $\sigma^2_{e_t}$ and $\sigma^2_{v_t}$ are non-Gaussian, owing to the log-transformation in the model for their evolution. MCMC can be implemented for these steps using, for example, Metropolis-Hastings moves. In our implementations we do not sample these parameters at the MCMC stage as they can be expected to move very slowly with time (the correct invariant distribution is of course maintained when these parameters are not moved by the MCMC sampler). When implementing MCMC steps with bridging densities, we will need to sample from a modified target density. We rewrite $p(x_t|x_{t-1}, y_t)$ in terms of the importance function and a weighting term:

$$p(x_t|x_{t-1}, y_t) \propto q(x_t|x_{t-1}, y_t) p(y_t|z_{t-1:t-P}, a_t, \sigma^2_{e_t}, \sigma^2_{v_t})$$

and then modify it to achieve the intermediate bridging densities:

$$\pi_m(x_t|x_{t-1}, y_t) \propto q(x_t|x_{t-1}, y_t) p(y_t|z_{t-1:t-P}, a_t, \sigma^2_{e_t}, \sigma^2_{v_t})^{1-\alpha_m}$$

In this way we can achieve a gradual transition from the original importance function ($\alpha_m = 1$) and the final target density ($\alpha_m = 0$). The full conditionals for Gibbs sampling are also modified for these intermediate densities. The full conditional for $z_t$ remains unchanged. The full conditional for $a_t$ is modified as follows:

$$\pi_m(a_t|z_t, \sigma^2_{e_t}, \sigma^2_{v_t}, x_{t-1}, y_t) = \mathcal{N}\left((\Phi_t^m)^{-1}\Theta_t^m, (\Phi_t^m)^{-1}\right) \tag{4.20}$$

$$\Phi_t^m = I_P/\sigma_a^2 + \frac{Z_{t-1}}{\sigma_{e_t}^2}\left((1-\alpha_m) + \frac{\alpha_m}{\phi_t \sigma_{e_t}^2}\right) \quad (4.21)$$

$$\Theta_t^m = \alpha a_{t-1}/\sigma_a^2 + z_{t-1:t-P} z_{t-1}/\sigma_{e_t}^2 - \alpha_m y_t z_{t-1:t-P}/(\sigma_{e_t}^2 + \sigma_{v_t}^2) \quad (4.22)$$

$$Z_{t-1} = z_{t-1:t-P} z_{t-1:t-P}^T \quad (4.23)$$

The required density ratio for the sampler with bridging densities is:

$$\pi_m(x_t|x_{t-1})/\pi_{m-1}(x_t|x_{t-1}) = p(y_t|z_{t-1:t-P}, a_t, \sigma_{e_t}^2, \sigma_{v_t}^2)^{\alpha_{m-1}-\alpha_m} \quad (4.24)$$

The MCMC algorithm with bridging densities can thus be summarised as

1. Set $t = 0$. Draw $x_0^{(i)} \sim p(x_0)$, $w_t^{(i)} = 1/N$, $i = 1, ..., N$
2. For $t = 1$ to $T$
   - $x_t^{(i)(0)} \sim q(x_t|x_{t-1}^{(i)}, y_t)$, $i = 1, ..., N$, see (4.11)
   - $w_t^{(i)(0)} = w_{t-1}^{(i)}$, $i = 1, ..., N$
   - For $m = 1$ to $M + 1$

     (Optional 1): Select with replacement from $\{x_t^{(i)(m-1)}, w_t^{(i)(m-1)}\}$

     $w_t^{(i)(m)} \propto w_t^{(i)(m-1)} \dfrac{\pi_m(x_t^{(i)(m-1)}|x_{t-1}^{(i)})}{\pi_{m-1}(x_t^{(i)(m-1)}|x_{t-1}^{(i)})}$, $i = 1, ..., N$, $\sum_{i=1}^{N} w_t^{(i)(m)} = 1$, see (4.24)

     $a_t^{(i)(m)} \sim \pi_m(a_t|z_t^{(i)(m-1)}, \sigma_{e_t}^{2(i)}, \sigma_{v_t}^{2(i)}, x_{t-1}^{(i)}, y_t)$, $i = 1, ..., N$, see (4.20)

     $z_t^{(i)(m)} \sim p(z_t|x_{t-1}^{(i)}, y_t, a_t^{(i)(m)}, \sigma_{e_t}^{2(i)}, \sigma_{v_t}^{2(i)})$, $i = 1, ..., N$, see (4.9)

     End
   - $w_t^{(i)} \propto w_t^{(i)(M+1)}$, $i = 1, ..., N$
   - (Optional 2) Select with replacement from $\{x_t^{(i)}\}$ with probability $\{w_t^{(i)}\}$

   End

This general algorithm includes both the techniques of (MacEachern et al. 1999) and (Gilks and Berzuini: this volume) as special cases. To achieve the (MacEachern et al. 1999) method, set $M = 0$ and do not perform the first (Optional 1) selection step; the second (Optional 2) selection step can be performed when the effective sample size or some other suitable measure of degeneracy becomes too low. To achieve the (Gilks and Berzuini: this volume) method set $M = 0$ and perform the first (Optional 1) selection step at every time $t$; do not perform the second (Optional 2) selection step. Notice that neither of these special cases incorporates the bridging density ideas, since with $M = 0$ there is no gradual transition from prediction density to target densities via intermediate bridging steps.

## 4.5 Simulation Results

We have tested all of the above algorithms extensively with the TVAR model. It is certainly possible to devise scenarios where none of the techniques works reliably, or where an impractically huge number of particles is required. One example of such a scenario is when the evolution variance $\sigma_a^2$ is set particularly low relative to the observation noise variance $\sigma_{v_t}^2$. In such cases the state trajectories very frequently diverge altogether from the true values and a large number of particles is essential. In other milder conditions all of the methods perform quite well with small particle cloud size.

In the simulations we aim to study several questions. The first is whether auxiliary particle filters, which are competitive in terms of computational load with the bootstrap filter, are a worthwhile modification for this class of models. Secondly, we explore the performance of the various MCMC-based procedures, aiming to discover whether these approaches, which can be considerably more demanding in computer power than the bootstrap filter, can give useful benefits in terms of robustness and accuracy of estimation. Within this class of MCMC algorithms we wish to explore the relative merits of the (MacEachern et al. 1999) approach, which does not involve a selection step before the MCMC moves, and the resample-move method of (Gilks and Berzuini: this volume), which always involves a selection step prior to the MCMC step. Finally, we wish to evaluate whether the use of bridging densities can aid estimation in a significant way for hostile environments. The simulations below go some way to answering these questions for the TVAR model class, although it is hard to draw any general conclusions for other models and data conditions not tested here.

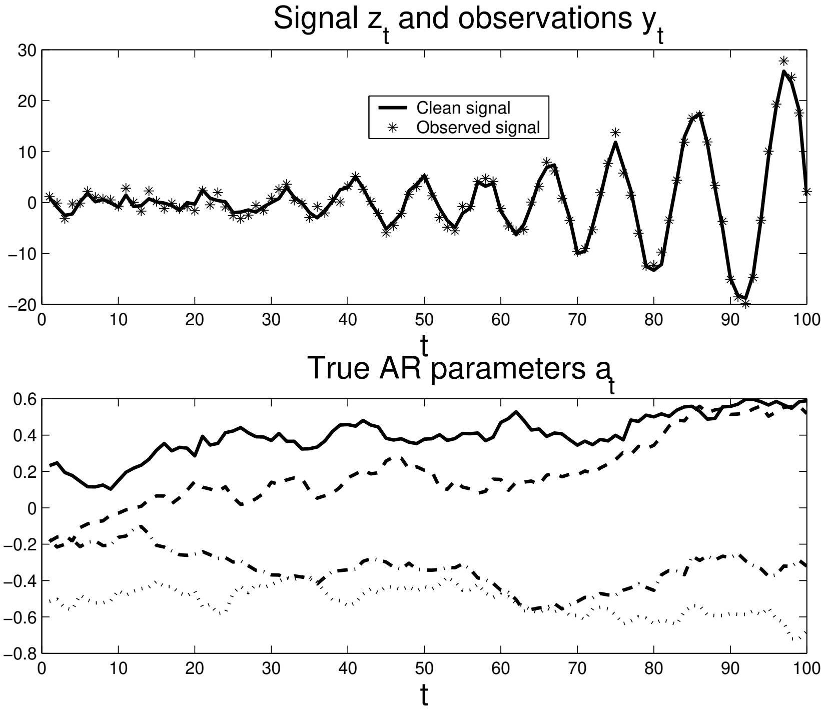
Experiments are carried out with multiple realisations of synthetic data. In this way we can go some way towards evaluating the expected performance of each of the methods in terms of mean-squared error. We estimate conditional means from the Monte Carlo output, focusing in particular upon the AR coefficients themselves, as follows:

$$E[a_t|y_{1:t}] \simeq \widehat{\mu}_t = \sum_{i=1}^{N} w_t^{(i)} a_t^{(i)}$$

Since the conditional mean is the minimum mean-squared error estimator for a parameter, we can evaluate the performance of each algorithm by averaging the squared error between $\widehat{\mu}_t$ and the true parameter value over many realisations of the data with the same hyperparameters.

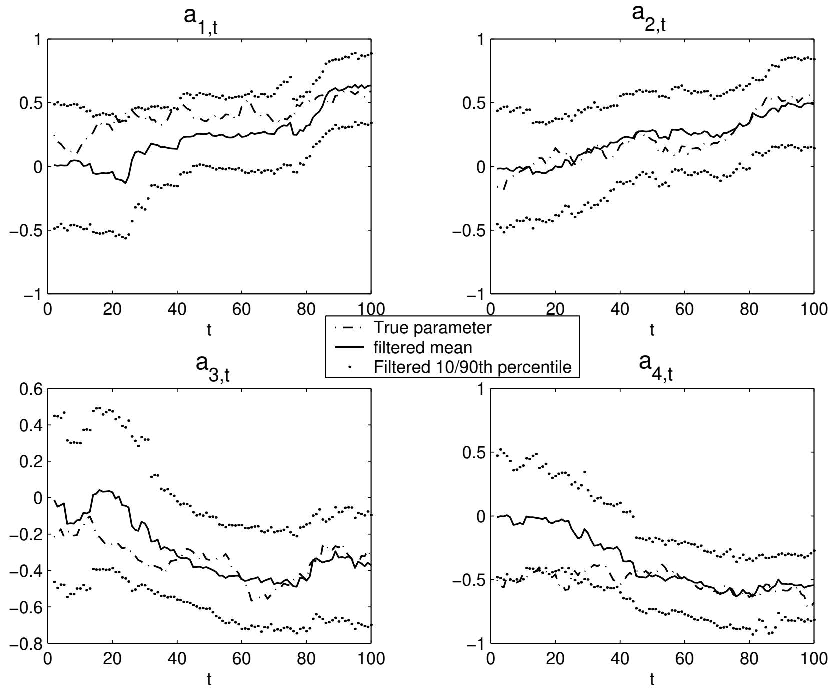
In the first set of simulations we compare the performance of the auxiliary particle filter (APF) with a basic bootstrap (SIR) filter. The simulations are carried out over 100 independent data realisations for a fourth order model ($P = 4$) with fixed hyperparameters. These parameters are given by

$\alpha = 0.996$, $\sigma_a = 0.03$ and $\sigma_{\phi_e} = 0.01$. Note that in these simulations the observation noise variance is fixed to $\sigma^2_{v_t} = 1$ and not estimated - it was found that estimation of this parameter sometimes led to divergence of the filter if the number of particles was low (i.e. the filter loses track entirely of the correct parameter trajectory and only regains it by chance). The remaining states are initialised randomly (both in generating the data realisations and in the subsequent filtering for $t = 0$) from an initial prior distribution $p(x_0)$ which is Gaussian and independent over all elements of $x_0$. Elements of $a_0$ and $z_{0:-P+1}$ each have variance 4 and mean 0, while $\phi_{e_0}$ has variance 1 and mean 1. A typical realisation of the AR coefficients and observed signal is shown in figure 4.5. In figure 4.5 some results are shown from the APF with $N = 1000$.




Typical signal and parameter trajectories.

It can be seen that the filter is able to track the coefficient variation with time quite well, although the percentiles show a great deal of uncertainty about the precise values. A comparative result is now given for the SIR filter and the APF. Both filters are run on the same 100 realisations of the data, with 500 particles. The mean-squared error is computed at each time point for all 100 realisations and the result is plotted in figure 4.5. As expected, the estimation error is much larger at the start of the data, and decreases once the filter has seen enough data to be sure of the parameter trajectory. The APF shows a small improvement in mean-squared error at all time points. This small but systematic improvement has been observed in nearly all of our trials, and so we conclude that the APF is a worthwhile modification in this class of mod-

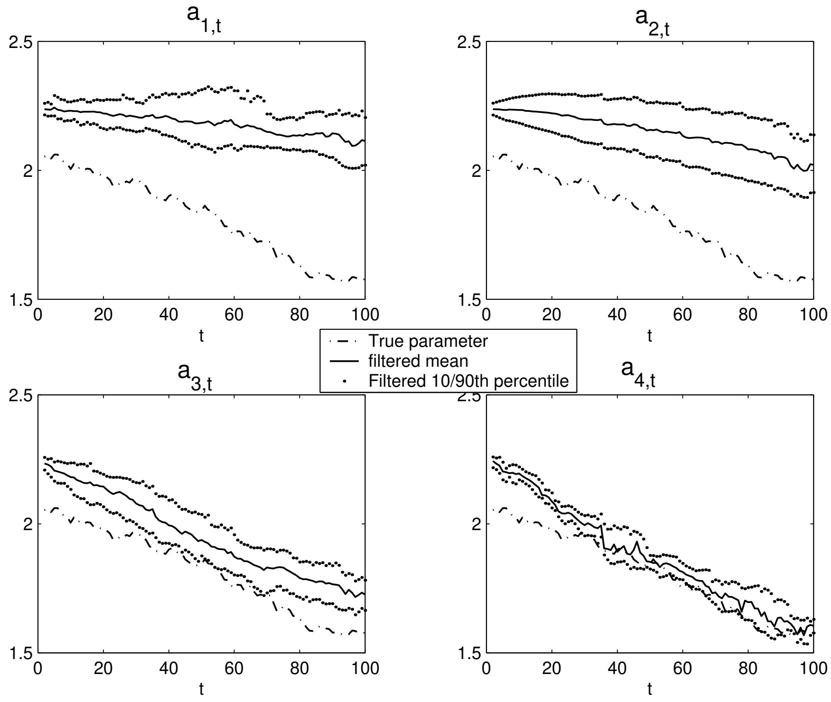



Filtering results using APF with $N = 1000$.

els. Furthermore, the APF has been found more robust to stability problems in the signal estimates than the SIR, which can occasionally generate large errors when the AR model is unstable. Next we consider the use of MCMC sampling steps in the filter, both with and without bridging densities. It should be remembered that the extra computational requirements for these procedures are very considerable, so that any improvements would have to be significant to justify their use. In simulations with artificially generated data of the type discussed so far, we have not found a significant inprovement in performance over the standard SIR or APF methods. However, the aim of the MCMC procedures is to robustify the filtering in challenging situations where the filtering density changes suddenly from one time point to the next. This can occur when outliers are present in the data, or step changes occur in the parameter trajectories. These situations very rarely arise when generating synthetic data from a known model, as above. However, they are much more likely to occur when working with real data, where unmodelled irregularities can occur in large datasets. It is in these situations that we can see the usefulness of the MCMC procedures. Quite a good illustration can be obtained by studying 'wrongly' initialised filters. Here the initial state probability is chosen to be quite distant from the true initial state and with a low variance. Hence the filter starts off along entirely the wrong trajectory, and it is expected that the likelihood at each successive time point will be very far from the prediction density. In the simulation performed here, the hyperparame-


Comparison of MSE values between APF and SIR ($N = 500$).

ters are $\alpha = 0.996$, $\sigma_a = 0.01$ and $\sigma_{\phi_e} = 0.01$. The smaller value of $\sigma_a$ means that the AR coefficients vary more slowly with time than before. The initial density for the coefficients has its mean offset by 0.1 from the initial density used to generate the data. Results of filtering for various schemes are shown in figure 4.5, plotting the first AR coefficient, $a_{1,t}$, which was typical of the performance for other parameters and other relisations of the data under the same conditions. The top two figures show that the APF is unable to track the parameter successfully, even when 10000 particles are used. The bottom left graph shows the standard MCMC method without bridging densities. Six MCMC iterations are used at each time step. This method is able to track the parameter trajectories quite successfully after 30-40 time steps, although it never quite succeeds in locking on to the true trajectory. The bottom right shows the use of MCMC with bridging densities. The bridging schedule is fixed throughout, with $M = 5$ and $\{\alpha_1, ..., \alpha_5\} = \{0.9, 0.7, 0.5, 0.3, 0.1\}$. The computational requirement is thus almost identical to that of the standard MCMC approach. The state trajectories lock on to the true path slightly more rapidly than MCMC, and this is borne out by the overall mean squared error figures of 0.0169 for the MCMC and 0.0085 for MCMC with bridging densities. In both cases, there was no selection step prior to carrying out the MCMC moves, i.e. the weights were accumulated right through to the end of the MCMC steps. Other simulations which included a selection step prior to the MCMC steps gave worse mean-squared error for this example, although it is hard to draw general conclusions from such a small-scale trial.




Comparison of algorithms for coefficient $a_{1,t}$.

# 5 Discussion

In this chapter we have discussed various methods for improving upon the basic particle filter, including a novel technique which employs bridging densities and MCMC moves. The application of the methods has been illustrated by a detailed case study of the time-varying autoregressive model. In simulations it can be concluded that the auxiliary particle filter gives a slight but systematic improvement in performance. The case for MCMC methods is less clear-cut, as they add significantly to the computational burden. However, we have demonstrated challenging cases where the MCMC techniques, both with and without bridging densities, can lead to dramatically better performance than SIR or APF. In our experiments with bridging densities we used a fixed schedule of annealing. However, there are open issues here of how a schedule should be chosen, and whether it can be adapted at each time step to the demands of the situation, for example how many MCMC steps should be applied and whether to use different numbers of steps for each particle. Further discussion and detail on these topics can be found in (Clapp 2000).

# 6 Acknowledgements

We wish to thank the anonymous referees for their constructive comments and suggestions, and also all those in the Signal Processing Group at Cambridge who have proof-read and commented on various stages of this work.


# References


Carpenter, J., Clifford, P. and Fearnhead, P. (1999). Building robust simulation-based filters for evolving datasets, *Tech. Rep. University of Oxford*.

Clapp, T. C. (2000). *Statistical Methods in the Processing of Communications Data*, PhD thesis, Cambridge University Engineering Department.

Clapp, T. C. and Godsill, S. J. (1999). Fixed-lag smoothing using sequential importance sampling, *in* J. Bernardo, J. Berger, A. Dawid and A. Smith (eds), *Bayesian Statistics VI*, Oxford University Press, pp. 743–752.

Doucet, A., Godsill, S. J. and Andrieu, C. (1999). On sequential Monte Carlo methods for Bayesian filtering, *Statistics and Computing*. Accepted for publication.

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

Godsill, S. J. (1997). Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes, *International Statistical Review* **65**(1): 1–21.

Godsill, S. J. and Rayner, P. J. W. (1998a). *Digital Audio Restoration: A Statistical Model-Based Approach*, Berlin: Springer, ISBN 3 540 76222 1.

Godsill, S. J. and Rayner, P. J. W. (1998b). Robust reconstruction and analysis of autoregressive signals in impulsive noise using the Gibbs sampler, *IEEE Trans. on Speech and Audio Processing* **6**(4): 352–372.

Godsill, S. J., Doucet, A. and West, M. (2000). Methodology for Monte Carlo smoothing with application to time-varying autoregressions, *Proc. International Symposium on Frontiers of Time Series Modelling*. Institute of Statistical Mathematics, Tokyo.

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

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

Kitagawa, G. and Gersch, W. (1996). *Smoothness Priors Analysis of Time Series, Lecture Notes in Statistics #116*, Springer-Verlag, New York.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations, *Journal of the American Statistical Association* **90**(430): 567–576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamical systems, *Journal of American Statistical Association* **93**: 1032–44.

MacEachern, S. N., Clyde, M. and Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: the next generation, *Can. J. Stats.* **27**: 251–267.

Neal, R. (1998). Annealed importance sampling, *Technical Report 9805*, Department of Statistics, University of Toronto.

Prado, R., West, M. and Krystal, A. (1999). Multi-channel EEG analyses via dynamic regression models with time-varying lag/lead structure, *ISDS Discussion paper 99-05, Duke University.*

Rajan, J. J., Rayner, P. J. W. and Godsill, S. J. (1997). A Bayesian approach to parameter estimation and interpolation of time-varying autoregressive processes using the Gibbs sampler, *IEE Proc. Vision, Image and Signal Processing.*

Shephard, N. (1994). Partial non-Gaussian state space, *Biometrika* **81**(1): 115–131.

Vermaak, J., Andrieu, C., Doucet, A. and Godsill, S. J. (1999). On-line Bayesian modelling and enhancement of speech signals, *Technical Report CUED/F-INFENG/TR.*, Cambridge University Engineering Department, Cambridge, England.

West, M. and Harrison, P. J. (1997). *Bayesian Forecasting and Dynamic Models*, 2nd edn, Springer-Verlag, New York.

West, M., Prado, R. and Krystal, A. (1999). Evaluation and comparison of EEG traces: Latent structure in non-stationary time series, *Journal of American Statistical Association* **94**: 1083–1095.