Monte Carlo Filtering of Piecewise Deterministic Processes

Nick Whiteley[*], Adam M. Johansen[†] and Simon Godsill [‡]

Abstract

We present efficient Monte Carlo algorithms for performing Bayesian inference in a broad class of models: those in which the distributions of interest may be represented by time marginals of continuous-time jump processes conditional on a realisation of some noisy observation sequence. The sequential nature of the proposed algorithm makes it particularly suitable for online estimation in time series. We demonstrate that two existing schemes can be interpreted as particular cases of the proposed method. Results are provided which illustrate significant performance improvements relative to existing methods. The appendix to this document can be found online.

keywords: Optimal Filtering, Particle Filters, Sequential Monte Carlo.

1 Introduction

Throughout statistics and related fields, there exist numerous problems which are most naturally addressed in a continuous time framework. A great deal of progress has been made over the past few decades in online estimation for partially-observed, discrete time series. This paper shows how to generalise standard discrete time *particle filtering* techniques to a broad class of continuous time models.

Within the Bayesian paradigm, the tasks of optimal filtering and smoothing correspond to obtaining, recursively in time, the posterior distribution of the trajectory of an unobserved stochastic process at a particular time instant, or over some interval, given a sequence of noisy observations made over time. State space models, in which the unobserved process is a

[*] Nick Whiteley is a Brunel Fellow in the Statistics Group, Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, UK. (email: nick.whiteley@bristol.ac.uk)

[†] Adam Johansen is an assistant professor in the Statistics Department, University of Warwick, Coventry, CV4 7AL, UK (email: a.m.johansen@warwick.ac.uk)

[‡] Simon Godsill is Professor of Statistical Signal Processing, Department of Engineering, University of Cambridge,Trumpington Street, Cambridge, CB2 1PZ, UK. (email: sjg@eng.cam.ac.uk)

discrete-time Markov chain, are very flexible and have found countless applications throughout statistics, engineering, biology, economics and physics. In many cases of interest the state-space model is nonlinear and nongaussian, and exact inference is intractable. Approximation methods must, therefore, be employed. Sequential Monte Carlo (SMC) methods (Cappé et al., 2007; Doucet and Johansen, 2009), approximate the sequence of posterior distributions by a collection of weighted samples, termed *particles*.

However, many physical phenomena are most naturally modelled in continuous time and this motivates the development of inference methodology for continuous time stochastic processes. Whilst it is possible to attempt discretisation, this can be computationally expensive and introduces a bias which is not easy to quantify. In this work we focus on estimating the trajectory of *piecewise deterministic* processes (PDP's), which evolve deterministically in continuous time *except* at a countable collection of stopping times at which they randomly *jump* (in the sense of potentially introducing a discontinuity into the trajectory) to a new value. PDP's were proposed by Davis (1984) and the connection between PDP's and marked point processes was investigated by Jacobsen (2006). Numerous problems can be straightforwardly and parsimoniously cast as filtering problems involving PDP's. We will present examples in the following two areas:

Mathematical Finance. Estimating the intensity of a shot noise Cox process (SNCP) plays a role in reinsurance and option pricing schemes. Exact inference for SNCP models is intractable. Approximate methods based on weak convergence to Gaussian processes in specific parameter regimes (Dassios and Jang, 2005) and Markov Chain Monte Carlo have been proposed (Centanni and Minozzo, 2006).

Object Tracking. It has been demonstrated that, in some object tracking scenarios, the trajectory of a manoeuvring object may be more parsimoniously modelled by a PDP than by traditional discrete-time models, see Lasdas and Davis (1989); Sworder et al. (1995); Godsill et al. (2007) and references therein.

The contribution of this paper is the development of efficient SMC algorithms for the filtering of any PDP. A key factor which determines the efficiency of SMC methods is the mechanism by which particles are propagated and re-weighted. If this mechanism is not carefully chosen, the particle approximation to the target distribution will be poor, with only a small subset of the particles having any significant weight. Over time, this in turn leads to loss of diversity in particle locations and instability of the algorithm. The proposed approach combats these issues, maintaining considerably more diversity in the particle system than previously-proposed approaches and therefore providing a better approximation to the target distribution. We are specifically interested in performing inference *online*, that is, as observations become available over time and in such a manner that computational complexity and storage requirements do not increase over time. This is essential in many applications.

In the next section, we specify the filtering model of interest more precisely. A representative example is presented and related existing works are discussed. In section 3 we describe the design of an SMC algorithm which is efficient for the processes of interest here. An auxiliary technique which can further improve the efficiency of the proposed method is also presented. We present results in section 4, for a SNCP model used in reinsurance pricing and an aircraft tracking model and demonstrate the improvement in performance over existing algorithms which is possible.

2 Background

Except where otherwise noted, distributions will be assumed to admit a density with respect to a suitable dominating measure, denoted dx, and, with a slight abuse of notation, the same symbol will be used to refer to both a measure and its associated density, so for example $\pi(dx) = \pi(x)dx$. We denote by $\delta_x(\cdot)$ the Dirac measure concentrated at x. For a Markov kernel $K(x, dy)$ acting from E_1 to E_2 and some probability measure $\mu(dx)$ on E_1, we will employ the following shorthand for the integral operation: $\mu K(\cdot) = \int_{E_1} K(x, \cdot)\mu(dx)$. When describing marginals of joint densities, we will use the notation of the following form: $\pi_n(x_k) = \int \pi_n(x_{1:n})\mathrm{d}x_{1:k-1}\mathrm{d}x_{k+1:n}$.

2.1 Model Specification

Consider first a pair Markov chain $(\tau_j, \theta_j)_{j\in\mathbb{N}}$, of non-decreasing times, $\tau_j \in \mathbb{R}^+$ and parameters, $\theta_j \in \Xi$ with transition kernel of the form:

$$p(d(\tau_j, \theta_j)|\tau_{j-1}, \theta_{j-1}) = f(d\tau_j|\tau_{j-1})q(d\theta_j|\theta_{j-1}, \tau_{j-1}, \tau_j). \tag{1}$$

There is no theoretical need to impose this conditional independence structure, but we do so here for ease of presentation. In some applications τ_j may be dependent upon θ_{j-1}, or the process $(\tau_j, \theta_j)_{j\in\mathbb{N}}$ may be non-Markovian. In principle the proposed methods can accommodate such dependence structures.

We next define a random continuous time counting process $(\nu_t)_{t\geq 0}$ as follows:

$$\nu_t = \sum_{j=1}^{\infty} \mathbb{I}_{[0,t]}(\tau_j) = \max\{j : \tau_j \leq t\}.$$

The *signal process*, $(\zeta_t)_{t\geq 0}$, which takes a value in Ξ at any time t and has known initial distribution, $\zeta_0 \sim q_0(\zeta_0)$, is then defined by: $\zeta_t = F(t, \tau_{\nu_t}, \theta_{\nu_t})$, with the conventions that $\tau_0 = 0$, $\theta_0 = \zeta_0$. The function $F : \mathbb{R}^+ \times \mathbb{R}^+ \times \Xi \to \Xi$, is deterministic, Ξ-valued and subject to the condition that $F(\tau_j, \tau_j, \theta_j) = \theta_j$, $\forall j \in \mathbb{N}$. From the initial condition ζ_0, a realisation of the signal process evolves deterministically according to F until the time of the first jump τ_1, at which time it takes the new value θ_1. The signal continues to evolve deterministically according to F until τ_2, at which time the signal acquires the new value θ_2, and so on.

We will be especially interested in the number of jumps occurring in the interval $[0, t_n]$ and therefore set $k_n \triangleq \nu_{t_n}$. Our model induces a joint prior distribution, $p_n(k_n, \tau_{1:k_n})$, on the number of jumps in $[0, t_n]$ and their locations:

$$p_n(k_n, d\tau_{1:k_n}) = S(t_n, \tau_{k_n}) \prod_{j=1}^{k_n} f(d\tau_j|\tau_{j-1}), \tag{2}$$

which has support on the disjoint union: $\bigcup_{k=0}^{\infty}\{k\} \times \mathbb{T}_{n,k}$, where $\mathbb{R}^k \supset \mathbb{T}_{n,k} = \{\tau_{1:k} : 0 < \tau_1 < \cdots < \tau_k \leq t_n\}$ and where $S(t, \tau)$ is the survivor function associated with the transition kernel $f(d\tau_j|\tau_{j-1})$:

$$S(t, \tau) = 1 - \int_{\tau}^{t} f(ds|\tau). \tag{3}$$

(3) is the conditional probability that, given the most recent jump occurred at τ, the next jump occurs after t. Thus (2) is the probability that there are precisely k_n jumps on $[0, t_n]$, with locations in infinitesimal neighbourhoods of $\tau_1, \tau_2, ..., \tau_{k_n}$.

The signal process is observed in the nth window $(t_{n-1}, t_n]$ through a collection of random variables, Y_n, conditionally independent of the past, given the signal process on $(t_{n-1}, t_n]$. Denote the likelihood function by $g(y_n|\zeta_{(t_{n-1},t_n]})$.

Given the function F, the path $(\zeta_t)_{t\in[0,t_n]}$ is completely specified by the initial condition ζ_0, the number of jumps, k_n, their locations $\tau_{1:k_n}$ and associated parameter values $\theta_{1:k_n}$. For each n, we will be interested in sampling such quantities. For each n, we therefore specify a distinct collection of random variables $X_n = (k_n, \zeta_{n,0}, \theta_{n,1:k_n}, \tau_{n,1:k_n})$, which takes its values in the disjoint union:

$$E_n = \bigcup_{k=0}^{\infty} \{k\} \times \Xi^{k+1} \times \mathbb{T}_{n,k},$$

Note that $E_n \subset E_{n+1}$. We also write $\zeta_{n,t} = F(t, \tau_{\nu_{n,t}}, \theta_{\nu_{n,t}})$.

In order to obtain the distribution of $(\zeta_t)_{t\in[0,t_n]}$, given the observations $y_{1:n}$, it would suffice to find $\pi_n(x_n)$, the posterior density of X_n, because, by construction, the signal process is a deterministic function of the jump times and parameters. This posterior, up to a constant of proportionality, has the form:

$$\pi_n(x_n) \propto p_n(k_n, \tau_{n,1:k_n}) q_0(\zeta_{n,0}) \prod_{j=1}^{k_n} q(\theta_{n,j}|\theta_{n,j-1}, \tau_{n,j}, \tau_{n,j-1}) \prod_{p=1}^{n} g(y_p|\zeta_{n,(t_{p-1},t_p]}). \tag{4}$$

As a filtering example, in order to obtain the distribution, $p(\zeta_{t_n}|y_{1:n})$, it suffices to obtain $\pi_n(\tau_{k_n}, \theta_{k_n})$. Exact inference for all non-trivial versions of this model is intractable and in section 3 we describe Monte Carlo approximation schemes.

2.2 A Motivating Example

Consider the planar motion of a vehicle which manoeuvres according to standard, piecewise constant acceleration dynamics. Each parameter may be decomposed into x and y components, each containing a position, s, velocity, u, and acceleration value, a,

$$\theta_j = \begin{bmatrix} \theta_j^x \\ \theta_j^y \end{bmatrix} \text{ and } F(t, \tau_{\nu_t}, \theta_{\nu_t}) = \begin{bmatrix} F^x(t, \tau_{\nu_t}, \theta_{\nu_t}) \\ F^y(t, \tau_{\nu_t}, \theta_{\nu_t}) \end{bmatrix}.$$

Here $\Xi = \mathbb{R}^6$ but the x and y components have identical parameters and evolutions; for brevity we describe only a single component: $\theta_j^x = [s_j^x \; u_j^x \; a_j^x]^T$ and,

$$F^x(t, \tau_{\nu_t}, \theta_{\nu_t}) = \begin{bmatrix} 1 & (t-\tau_{\nu_t}) & \frac{1}{2}(t-\tau_{\nu_t})^2 \\ 0 & 1 & (t-\tau_{\nu_t}) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{\nu_t}^x \\ u_{\nu_t}^x \\ a_{\nu_t}^x \end{bmatrix}.$$

At time zero the vehicle has position, velocity and acceleration ζ_0. At τ_j, the acceleration of the vehicle jumps to a new, random value according to $q(d\theta_j|\theta_{j-1}, \tau_{j-1}, \tau_j)$, which is specified by:

$$\begin{aligned} q(d\theta_j^x|\theta_{j-1}, \tau_j, \tau_{j-1}) &= \delta_{s_j^{x,-}}(ds_j^x)\delta_{u_j^{x,-}}(du_j^x) q(da_j^x), \\ s_j^{x,-} &= s_{j-1}^x + u_{j-1}^x(\tau_j - \tau_{j-1}) + a_{j-1}^x(\tau_j - \tau_{j-1})^2/2, \\ u_j^{x,-} &= u_{j-1}^x + a_{j-1}^x(\tau_j - \tau_{j-1}). \end{aligned}$$

The component of F in the y-direction is equivalent. This model is considered a suitable candidate for the benchmark fighter-aircraft trajectory as described in Blair et al. (1998), shown in figure 3a. As an example observation model, at each time t_n the Cartesian position of the vehicle is observed through some noisy sensor. An example inference task is then to recursively estimate the position of the vehicle, and the time of the most recent jump in its acceleration, given observations from the noisy sensor. We return to this example in the sequel.

2.3 Related Work

Inference schemes based upon direct extension of the particle filter have been devised for the process of interest. The variable rate particle filter (VRPF) of Godsill and Vermaak (2004); Godsill et al. (2007) is one such scheme. Algorithms for filtering of Brownian motion–driven diffusions were developed in Godsill (2007). A related method was presented independently in Maskell (2004). The relationship between the proposed method and these existing algorithms is made precise in the sequel.

In the standard discrete-time particle filtering context, it is well known that sampling from the prior distribution can be inefficient, yielding importance weights of high variance. This problem is exacerbated when constructing SMC algorithms for PDP's.

Resampling, which is an essential component of SMC algorithms for online filtering, results in multiple copies of some particles. Under the prior distribution for the class of models considered here, there is significant positive probability that zero jumps will occur between one observation time and the next. Proposing from the prior distribution after resampling can therefore result in multiple copies of some particles being propagated over several iterations of the algorithm, without diversification. This phenomenon is most obvious when the expected jump arrival rate is low relative to the rate at which observations are made (as is the case in applications of interest). This in turn leads to the accumulation of errors in the particle approximation to the target distribution and instability of the algorithm. Computational methods to avoid repeated calculations and diversification by 'state regeneration' were described in Godsill et al. (2007), but these do not address the underlying problem and involve proposing particles without taking into account new observations.

Finally, we note that Chopin (2006) addressed a related but discrete-time change-point problem, which propagates the posterior distribution for the time since the most recent change-point. Chopin remarks that slow-mixing properties make the inclusion of MCMC diversification moves (Gilks and Berzuini (2001)) in the accompanying SMC algorithm essential for efficient operation. The 'adjustment' moves proposed below address the same basic problem within the framework of the present paper and diversification via MCMC moves is also possible.

3 Methodology

The SMC samplers framework of Del Moral et al. (2006) is a general method for obtaining a set of samples from a sequence of distributions which are defined on the same or different spaces. This generalisation of the standard SMC method (in which the object distributions exist on spaces of strictly increasing dimension) has recently been applied to trans-dimensional problems (Doucet et al., 2006).

Given a collection of spaces $(E_n)_{n\in\mathbb{N}}$, upon which the sequence of probability measures from which we wish to sample, $(\pi_n)_{n\in\mathbb{N}}$ is defined, the SMC sampler approach is to construct a sequence of distributions $(\widetilde{\pi}_n)_{n\in\mathbb{N}}$ upon the sequence of spaces $\left(\prod_{p=1}^{n} E_p\right)_{n\in\mathbb{N}}$, which have the target at time index n as a marginal distribution at that time index. The synthetic distributions are (assuming that a density representation exists) defined by:

$$\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{p=1}^{n-1} L_p\left(x_{p+1}, x_p\right), \tag{5}$$

where $(L_n)_{n\in\mathbb{N}}$ is a sequence of 'backward' Markov kernels with L_n acting *from* E_{n+1} *into* E_n. With this structure, an importance sample from $\widetilde{\pi}_n$ is obtained by taking the path $x_{1:n-1}$, a sample from $\widetilde{\pi}_{n-1}$, and extending it with a Markov kernel, K_n, which acts from E_{n-1} into E_n, providing samples from $\widetilde{\pi}_{n-1} \times K_n$ and leading to the incremental importance weight:

$$w_n(x_{n-1}, x_n) \propto \frac{\widetilde{\pi}_n(x_{1:n})}{\widetilde{\pi}_{n-1}(x_{1:n-1})K_n(x_{n-1}, x_n)} = \frac{\pi_n(x_n)L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}.$$

Remark 1. *Although, for simplicity, we have above followed the convention in the literature and assumed that all distributions and kernels of interest may be described by a density with respect to some common dominating measure, this is not necessarily the case. All that is actually required is that the product distribution defined by $\pi_n \otimes L_{n-1}(d(x_n, x_{n-1}))$ is dominated by $\pi_{n-1} \otimes K_n(d(x_{n-1}, x_n))$ and the importance weight is then the corresponding Radon-Nikodým derivative. Thus the backward kernel L_{n-1} must be chosen such that this requirement is met.*

Under this approach the importance weights at time index n depend only upon x_n and

x_{n-1}. This means the full history of the sampler need not be stored. In the context of the filtering problem under consideration, recall that x_n actually specifies a path of $(\zeta_t)_{t\in[0,t_n]}$, but we will construct samplers with importance weights which depend only upon some recent history of x_n and x_{n-1}, yielding a scheme with storage requirements which do not increase over time.

3.1 PDP Particle Filter

We next describe the design of a SMC scheme to target the distributions of interest in our filtering problem. By applying the SMC samplers method to the sequence of distributions $(\pi_n(x_n))_{n\in\mathbb{N}}$, see (4), we can obtain recursive schemes which propagate particle approximations to the distributions of various quantities in the recent history of the signal process. As a simple example this allows us to maintain an approximation to each marginal distribution $\pi_n(\tau_{k_n}, \theta_{k_n})$, and thus to $p(\zeta_{t_n}|y_{1:n})$. From the set of N particles, $\{(k_n\ \tau_{k_n}\ \theta_{k_n})^{(i)}\ ,\ w_n^{(i)}\}_{i=1}^N$, the filtering distribution for the signal process can be approximated by:

$$p^N(d\zeta_{t_n}|y_{1:n}) = \sum_{i=1}^N w_n^{(i)} \delta_{\zeta_{t_n}^{(i)}}(d\zeta_{t_n}), \qquad \zeta_{t_n}^{(i)} = F\left(t_n, \tau^{(i)}_{k_n^{(i)}}, \theta^{(i)}_{k_n^{(i)}}\right).$$

We next give specific details of how such algorithms can be constructed. The explicit treatment of the dimensionality of the problem gives us control over the proposal of different numbers of jumps. Furthermore, the SMC samplers framework accommodates a more efficient proposal mechanism than that of the VRPF by permitting 'adjustment' moves as described below.

3.1.1 Choice of Kernels

In addition to the choice of the proposal kernels K_n, it is necessary to select a sequence of backward kernels L_n. The appearance of these kernels in the weight expression makes it clear that it will be extremely important to select these carefully. In the proposed algorithms we will be employing proposal kernels which are mixtures, for example each mixture component applying a different increment to the dimensionality parameter k_n. In general it is valid to allow the mixture weights to be a function of the current state, for example:

$$K_n(x_{n-1}, x_n) = \sum_{m=1}^M \alpha_{n,m}(x_{n-1}) K_{n,m}(x_{n-1}, x_n), \qquad \forall x_n,\ \sum_{m=1}^M \alpha_{n,m} = 1.$$

Del Moral et al. (2006) provide an expression for the optimal backward kernel for such a proposal kernel (in the sense that the variance of the importance weights is minimized if resampling is conducted at every step). When this optimal backward kernel is employed, the importance weight becomes:

$$w_n(x_{n-1}, x_n) \propto \frac{\pi_n(x_n)}{\int_{E_{n-1}} \pi_{n-1}(x_{n-1}) \left[\sum_{m=1}^{M} \alpha_{n,m}(x_{n-1}) K_{n,m}(x_{n-1}, x_n)\right] \mathrm{d}x_{n-1}}. \tag{6}$$

Whenever it is possible to use the optimal backward kernel, one should do so. However, in most instances the integral in the denominator of (6) will prove intractable, and an approximation to the optimal kernel must be used instead. We will consider backward kernels of the mixture form

$$L_{n-1}(x_n, x_{n-1}) = \sum_{m=1}^{M} \beta_{n-1,m}(x_n) L_{n-1,m}(x_n, x_{n-1}).$$

In the context of mixture proposal kernels, it was suggested in Del Moral et al. (2006), that IS could be performed on a higher dimensional space, involving the space of mixture component indicators, in which case the incremental importance weight is specified by:

$$w_n(x_{n-1}, m, x_n) \propto \frac{\pi_n(x_n)\beta_{n-1,m}(x_n)L_{n-1,m}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1})\alpha_{n,m}(x_{n-1})K_{n,m}(x_{n-1}, x_n)}. \tag{7}$$

The design of the proposal kernel also plays a significant role in the performance of the algorithm. In order to minimise the variance of the importance weights, it must be well matched to the target distribution and hence to the observations. There are many different proposal kernels which could be employed in the context of PDP filtering. In order to provide some guidance, we next show how to build a generic kernel, consisting of two moves, in order to highlight issues which affect the algorithm's performance. An algorithm built around the below moves and possibly also involving the application of a Metropolis-Hastings kernel (further details below), constitutes a general strategy for the filtering PDPs. Another example can be found the appendix, where the algorithm of Maskell (2004) and the VRPF of Godsill et al. (2007) are specified in the PDP particle filter framework.

3.1.2 Standard Moves

We here consider two generic moves, an *adjustment* move and a *birth* move.

Adjustment Move $K_{n,\mathrm{a}}$. The dimensionality is maintained and the most recent jump time

$\tau_{n-1,k_{n-1}}$, is replaced by a draw from a distribution $h_n(\cdot|x_{n-1})$ with support $(\tau_{n-1,k_{n-1}-1}, t_n]$, yielding a new jump time τ_{n,k_n}.

$$K_{n,\mathrm{a}}(x_{n-1}, dx_n) = \delta_{k_{n-1}}(k_n)\delta_{\tau_{n-1,1:k_{n-1}-1}}(d\tau_{n,1:k_n-1})\delta_{\theta_{n-1,1:k_{n-1}}}(d\theta_{n,1:k_n})h_n(d\tau_{n,k_n}|x_{n-1}). \quad (8)$$

It can be shown that the optimal choice (in the sense of minimizing the conditional variance of the importance weight, given that an adjustment move is to be made and that the optimal backwards kernel is employed) of $h_n(\cdot|x_{n-1})$ is the full conditional distribution $\pi_n(\cdot|x_n \setminus \tau_{n,k_n})$, where $x_n \setminus \tau_{n,k_n}$ denotes all components of x_n other than τ_{n,k_n}. A similar adjustment move can be devised to obtain a new value for the most recent parameter θ_{n,k_n}.

Birth Move $K_{n,\mathrm{b}}$. The dimensionality is increased and additional jump times and parameters are sampled. In a simple case, the number of jumps is incremented, $k_n = k_{n-1} + 1$, a new jump, τ_{n,k_n} is proposed from a distribution $h_n(\cdot|\tau_{n-1,k_{n-1}})$ on $(\tau_{n-1,k_{n-1}}, t_n]$, and a new parameter is then drawn from a proposal distribution $\eta_n(\cdot|x_n \setminus \theta_{n,k_n})$:

$$\begin{aligned} K_{n,\mathrm{b}}(x_{n-1}, dx_n) = {} & \delta_{k_{n-1}+1}(k_n)\delta_{\tau_{n-1,1:k_{n-1}}}(d\tau_{n,1:k_n-1}) \\ & \times \delta_{\theta_{n-1,1:k_{n-1}}}(d\theta_{n,1:k_n-1})h_n(d\tau_{n,k_n}|\tau_{n-1,k_{n-1}})\eta_n(d\theta_{n,k_n}|x_n \setminus \theta_{n,k_n}). \quad (9) \end{aligned}$$

It can be shown that the optimal choice of $\eta_n(\cdot|x_n \setminus \theta_{n,k_n})$ is the full conditional $\pi_n(\theta_{n,k_n}|x_n \setminus \theta_{n,k_n})$, given that a birth move is to be made, the optimal backwards kernel employed and τ_{n,k_n}. If $\tau_{n,k_n} \leq t_{n-1}$ this amounts to altering the trajectory $(\zeta_t)_{t\in[\tau_{n,k_n}, t_{n-1}]}$ and extending the trajectory onto $(t_{n-1}, t_n]$.

It should be noted that, due to the nested structure of the sequence $(E_n)_{n\in\mathbb{N}}$ and the support of the sequence $(\pi_n)_{n\in\mathbb{N}}$, there is no technical requirement to include a dimensionality reducing component, or *death* move, in the proposal. It is technically possible to do so, although the corresponding optimal backward kernel is rarely available exactly. Birth/death moves can also be applied via a trans–dimensional MCMC kernel.

Design of proposal kernels. The full conditional distributions for the most recent jump time and associated parameter are respectively given by:

$$\begin{aligned} \pi_n(\tau_{n,k_n}|x_n \setminus \tau_{n,k_n}) \propto {} & S(t_n, \tau_{n,k_n})f(\tau_{n,k_n}|\tau_{n,k_n-1})q(\theta_{n,k_n}|\theta_{n,k_n-1}, \tau_{n,k_n}, \tau_{n,k_n-1}) \\ & \cdot \prod_{p=Q_n}^{n} g(y_p|\zeta_{n,(t_{p-1},t_p]}) \cdot \mathbb{I}_{[\tau_{n,k_n-1},t_n]}(\tau_{n,k_n}), \quad (10) \end{aligned}$$

$$\pi_n(\theta_{n,k_n}|x_n \setminus \theta_{n,k_n}) \propto q(\theta_{n,k_n}|\theta_{n,k_{n-1}}, \tau_{n,k_n}, \tau_{n,k_n-1}) \prod_{p=Q'_n}^{n} g(y_p|\zeta_{n,(t_{p-1},t_p]}), \quad (11)$$

where $Q_n = \inf\{p : t_p \geq \tau_{n,k_n-1}\}$ and $Q'_n = \inf\{p : t_p \geq \tau_{n,k_n}\}$. When the full conditional distributions are not available analytically, sensible approximations should be employed. We note that such approximations do not affect the *exactness* of the algorithm; just the estimator variance.

One possible approach to the adjustment move is to propose by adding a small random perturbation to the existing most recent jump time. This technique is applied in section 4.2. An alternative, but more expensive strategy would be to obtain a piecewise linear approximation of (10) by evaluating the right hand side of (10) on a grid of points in $(\tau_{n,k_n-1}, t_n]$, interpolating and normalizing. Methods to approximate (11) will usually be model–specific. To provide guidance, we next give an illustrative example. Consider the case in which, a–priori, the parameters evolve according to

$$\theta_{n,k_n} = F(\tau_{n,k_n}, \tau_{n,k_n-1}, \theta_{n,k_n-1}) + U_n,$$

where U_n is an independent disturbance of known distribution and where observations are made point–wise, i.e.,

$$Y_n = G(\zeta_{t_n}) + V_n = G(F(t_n, \tau_{n,k_n}, \theta_{n,k_n})) + V_n,$$

where G is a possibly nonlinear observation function and V_n is an independent noise disturbance of known distribution. For (11), it may be possible to obtain a tractable approximation of the likelihood terms by local–linearisation of $G \circ F$. The same approach may be employed to approximate $q(\theta_{n,k_n}|\theta_{n,k_n-1}, \tau_{n,k_n}, \tau_{n,k_n-1})$ and hence the full conditional distribution. Approximations using this approach were applied in Whiteley et al. (2007).

Backward Kernel and Importance Weights. Given a proposal kernel consisting of one or more moves, we need a strategy for designing a corresponding backward kernel. We adopt the approach as in Del Moral et al. (2006) and first design one backward kernel component corresponding to each proposal kernel component, and then combine the backward kernel components.

Sensible approximations of the optimal backwards kernel must be employed, whilst ensuring that the requirement described in remark (1) is met. For the forward kernel components described above, we advocate the use of the following backward kernel components or approximations thereof:

$$L_{n-1,\mathrm{a}}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1})K_{n,\mathrm{a}}(x_{n-1}, x_n)}{\pi_{n-1}K_{n,\mathrm{a}}(x_n)}, \quad L_{n-1,\mathrm{b}}(x_n, x_{n-1}) = \frac{\pi_{n-1}(x_{n-1})K_{n,\mathrm{b}}(x_{n-1}, x_n)}{\pi_{n-1}K_{n,\mathrm{b}}(x_n)}.$$

We then need a way to combine these backward kernel components. For the two move types above, a strategy which has been found to work in practice is simply to set, for all m, $\beta_{n-1,m}(x_n) = 1/M$, where M is the total number of components in the proposal kernel. This approach can be expected to perform well when there is little overlap in support of $\pi_{n-1}K_{n,\mathrm{b}}$ and $\pi_{n-1}K_{n,\mathrm{a}}$. Then, using the expression in (7), the corresponding importance weights are given by the following expressions.

For the adjustment move applied to the most recent jump time τ_{n,k_n}, employing the optimal proposal distribution, the incremental weight is given by:

$$\begin{aligned} w_n(x_{n-1}, x_n) &\propto \frac{\pi_n(x_n)\pi_{n-1}(\theta_{n-1,k_{n-1}}|x_{n-1} \setminus \theta_{n-1,k_{n-1}})}{\alpha_{n,\mathrm{a}}(x_{n-1})\pi_{n-1}(x_{n-1})\pi_n(\theta_{n,k_n}|x_n \setminus \theta_{n,k_n})} \\ &= \frac{1}{\alpha_{n,\mathrm{a}}(x_{n-1})} \frac{S(t_n, \tau_{n,k_n})}{S(t_{n-1}, \tau_{n-1,k_{n-1}})} \frac{f(\tau_{n,k_n}|\tau_{n,k_n-1})}{f(\tau_{n-1,k_{n-1}}|\tau_{n-1,k_{n-1}-1})} \\ &\cdot \frac{q(\theta_{n,k_n}|\theta_{n,k_n-1}, \tau_{n,k_n}, \tau_{n,k_n-1})}{q(\theta_{n-1,k_{n-1}}|\theta_{n-1,k_{n-1}-1}, \tau_{n-1,k_{n-1}}, \tau_{n-1,k_{n-1}-1})} \\ &\cdot \frac{\pi_{n-1}(\tau_{n-1,k_{n-1}}|x_n \setminus \tau_{n-1,k_{n-1}})}{\pi_n(\tau_{n,k_n}|x_{n-1} \setminus \tau_{n,k_n})} \frac{\prod_{p=R_n}^{n} g(y_p|\zeta_{n,(t_{p-1},t_p]})}{\prod_{p=R_n}^{n-1} g(y_p|\zeta_{n-1,(t_{p-1},t_p]})}, \end{aligned} \tag{12}$$

where $R_n = \inf\{p : t_p \geq (\tau_{n,k_n} \wedge \tau_{n-1,k_{n-1}})\}$ and with the convention $\prod_{p=n}^{n-1} g(y_p|\zeta_{n-1,(t_{p-1},t_p]}) = 1$. For the birth move, employing the optimal proposal distribution,

$$\begin{aligned} w_n(x_{n-1}, x_n) &\propto \frac{\pi_n(x_n)}{\alpha_{n,\mathrm{b}}(x_{n-1})\pi_{n-1}(x_{n-1})h_n(\tau_{n,k_n}|\tau_{n-1,k_{n-1}})\pi_n(\theta_{n,k_n}|x_n \setminus \theta_{n,k_n})} \\ &= \frac{1}{\alpha_{n,\mathrm{b}}(x_{n-1})} \frac{S(t_n, \tau_{n,k_n})}{S(t_{n-1}, \tau_{n-1,k_{n-1}})} \frac{f(\tau_{n,k_n}|\tau_{n,k_n-1})}{h_n(\tau_{n,k_n}|\tau_{n-1,k_{n-1}})} \\ &\cdot \frac{q(\theta_{n,k_n}|\theta_{n,k_n-1}, \tau_{n,k_n}, \tau_{n,k_n-1})}{\pi_n(\theta_{n,k_n}|x_n \setminus \theta_{n,k_n})} \frac{\prod_{p=R'_n}^{n} g(y_p|\zeta_{n,(t_{p-1},t_p]})}{\prod_{p=R'_n}^{n-1} g(y_p|\zeta_{n-1,(t_{p-1},t_p]})}, \end{aligned} \tag{13}$$

where $R'_n = \inf\{p : t_p \geq \tau_{n,k_n})\}$ and with the convention $\prod_{p=n}^{n-1} g(y_p|\zeta_{n-1,(t_{p-1},t_p]}) = 1$. When other proposal distributions are used, the incremental weights are of the same form, but with the proposal distributions employed substituted for the corresponding full conditionals in (12) and (13).

Proposal Mixture Weights. If the proposal mixture weights, $\alpha_{n,\mathrm{a}}(x_{n-1})$ and $\alpha_{n,\mathrm{b}}(x_{n-1})$ are chosen to depend only on a recent history of x_{n-1}, then the incremental weights (12) and (13) depend only upon a recent history of x_n. A simple choice of these proposal mixture weights is in terms of prior probabilities. If employing a kernel which consists of a birth move and an adjustment move, we could define the mixture weight for the adjustment, $\alpha_{n,\mathrm{a}}(x_{n-1})$, move as follows:

$$\alpha_{n,\mathrm{a}}(x_{n-1}) = S(t_n, \tau_{n-1,k_{n-1}}), \tag{14}$$

which is the conditional prior probability of zero jumps in $(\tau_{n-1,k_{n-1}}, t_n]$, given $\tau_{n-1,k_{n-1}}$. Ideally we would like to take into account information from the observations in order to adapt the mixture weights and minimize the variance of the importance weights. Unfortunately this is usually intractable in practice.

3.2 Computational Issues and Further Considerations

In order for the computational cost of evaluating the incremental importance weights to be bounded uniformally over iterations of the algorithm, it is necessary to bound the random number of likelihood terms which need to be evaluated in (12) and (13) (including implicitly the full conditional distributions, (10) and (11), if they are employed). This can be achieved by imposing some constraints on the proposal mixture weights and proposal distributions. To bound the cost of evaluating (12), the proposal mixture weights can be chosen such that an adjustment move is never applied when $t_n - \tau_{n-1,k_{n-1}-1}$ exceeds some threshold, e.g.,

$$\alpha_{n,\mathrm{a}}(x_{n-1}) = \begin{cases} 0 & \text{if } t_n - \tau_{n-1,k_{n-1}-1} > t_{\max}, \\ S(t_n, \tau_{n-1,k_{n-1}}) & \text{otherwise,} \end{cases}$$

where $t_{\max}$ is chosen by the user. In order for the computational cost of evaluating (13) to be bounded, the support of $h_n(\cdot|\tau_{n-1,k_{n-1}})$ could be restricted to $(t_n - t_{\max}, t_n] \cap (\tau_{n-1,k_{n-1}}, t_n]$ whenever $t_n - \tau_{n-1,k_{n-1}-1} > t_{\max}$. Under this strategy, and on any event, the number of likelihood evaluations in the importance weights and full conditionals is bounded by a deterministic constant which depends on $t_{\max}$ but not on n. In practice, $t_{\max}$ can be chosen to be large enough so that the event such that $t_n - \tau_{n-1,k_{n-1}} > t_{\max}$ rarely occurs.

A technical requirement of importance sampling schemes is that support of the proposal distribution includes that of the posterior distribution. It appears never to have been mentioned in the literature that in the context of trans-dimensional inference in time series, this imposes the requirement that a forward kernel capable of proposing any positive number of births in the interval $(t_{n-1}, t_n]$ must be employed if there is a positive probability associated with such configurations under the target distribution. In principle it might be sufficient to employ a birth move which introduces $1+B$ new jumps, with B a Poisson random variable of

Algorithm 1 PDP Particle Filter

```
1: $n = 1$
2: for $i = 1$ to $N$ do
3:     $X_1^{(i)} \sim \eta(\cdot)$        {where $\eta(x)$ is an instrumental distribution.}
4:     $W_1^{(i)} \propto \frac{\pi_1(X_1^{(i)})}{\eta(X_1^{(i)})}$
5: end for
6: $n \leftarrow n + 1$
7: for $i = 1$ to $N$ do
8:     $X_n^{(i)} \sim K_n(X_{n-1}^{(i)}, \cdot)$
9:     $W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(X_n^{(i)}) L_{n-1}(X_n^{(i)}, X_{n-1}^{(i)})}{\pi_{n-1}(X_{n-1}^{(i)}) K_n(X_{n-1}^{(i)}, X_n^{(i)})}$
10: end for
11: Resampling can be conducted at this stage.
12: Optionally, move each $X_n^{(i)}$ according to a $\pi_n$-invariant Markov kernel.
13: goto 6
```

low intensity, to ensure convergence (asymptotically in the number of particles) to the target distribution. In practice, for finite samples, if the intensity is small enough it is equivalent to never proposing more than a single new jump.

Resampling should be carried out after any iteration which causes the effective sample size (ESS), Kong et al. (1994), to fall below a reasonable threshold (typically around half of the total number of particles), to prevent the sample becoming degenerate with a small number of samples having very large weights. Post–resampling, the importance weights are set to be uniform. After resampling at the nth iteration, a Markov kernel of invariant distribution π_n, for example a reversible jump Metropolis-Hastings kernel, can be applied to the particle system.

The generic scheme for a PDP particle filter is given in algorithm 1. As is demonstrated in the examples, inserting a mixture of the moves as described in 3.1.2 and the specified weight expressions into this algorithm provides a generic technique for the filtering of PDPs.

3.3 Auxiliary Methods

The auxiliary particle filter of Pitt and Shephard (1999) introduces a set of auxiliary particle weights before the resampling step, with aim of 'pre-selecting' particles so as to reduce the variance of the importance weights at the next time step. In Johansen and Doucet (2008), the auxiliary method was reinterpreted as a standard sequential importance resampling algorithm targeting a sequence of auxiliary distributions, which themselves form the proposal distributions of an IS scheme. The generalisation from particle filters to general SMC samplers has also been made in Johansen and Doucet (2007).

The idea is to use the observation at time $n + 1$ to guide the weighting of the particle set at time n so that resampling does not eliminate particles which will be more favourably weighted once that observation has been taken into account. This pre-weighting is then corrected for by importance weighting after the resampling step. An auxiliary version of the algorithm presented above amounts to an SMC sampler targeting an auxiliary sequence of distributions on $(E_n)_{n\geq 0}$, which will be denoted by $(\mu_n)_{n\geq 1}$, and a sequence of importance weights, $(\widetilde{W}_n)$ which correct for the discrepancy between $(\mu_n)_{n\geq 1}$ and $(\pi_n)_{n\geq 1}$.

We will focus on auxiliary distributions of the following form:

$$\mu_n(x_n) \propto V_n(\tau_{n,k_n}, \theta_{n,k_n}, y_{n+1})\pi_n(x_n)$$

where for each n, $V_n : \mathbb{R}^+ \times \Xi \rightarrow (0, \infty)$ is a potential function, i.e. the algorithm will pre-select particles on the basis of the most recent jump time and associated parameter and their interaction with the next observation. It is important to choose the potential function V_n sensibly, guidance is provided in Pitt and Shephard (1999) and Johansen and Doucet (2007) taking into account recent developments. An example V_n is presented in section 4.2.

The application of the auxiliary method to the example sampler above is described in Algorithm 2. Steps 5 and 13 in this algorithm indicate where the proposed scheme yields particle sets targeting the distributions of interest, $(\pi_n)_{n\in\mathbb{N}}$, in the sense of approximating expectations with respect to these distributions. The theoretical properties of the proposed

Algorithm 2 Auxiliary PDP Particle Filter

```
1:  $n = 1$
2:  for $i = 1$ to $N$ do
3:      $X_1^{(i)} \sim \eta(\cdot)$      {where $\eta(x)$ is an instrumental distribution.}
4:      $\widetilde{W}_1(X_1^{(i)}) \propto \frac{\pi_1(X_1^{(i)})}{\eta(X_1^{(i)})}$
5:      $\int \varphi(x_1)\pi_n(x_1)\mathrm{d}x_1 \approx \frac{\sum_{i=1}^N \widetilde{W}_1(X_1^{(i)})\varphi_n(X_n^{(i)})}{\sum_{i=1}^N \widetilde{W}_1(X_1^{(i)})}$
6:      $W_1(X_1^{(i)}) \propto V_1(\tau_{1,k_1}^{(i)}, \theta_{1,k_1}^{(i)}, y_2)\widetilde{W}_1(X_1^{(i)})$
7:  end for
8:  $n \leftarrow n + 1$
9:  Resample from distribution defined by $\{W_{n-1}^{(i)}\}_{i=1}^N$
10: for $i = 1$ to $N$ do
11:     $X_n^{(i)} \sim K_n(X_{n-1}^{(i)}, \cdot)$
12:     $\widetilde{W}_n^{(i)}(X_{n-1}^{(i)}, X_n^{(i)}) \propto \frac{1}{V_{n-1}(\tau^{(i)}_{n-1,k^{(i)}_{n-1}}, \theta^{(i)}_{n-1,k^{(i)}_{n-1}}, y_n)} \frac{\pi_n(X_n^{(i)})L_{n-1}(X_n^{(i)}, X_{n-1}^{(i)})}{\pi_{n-1}(X_{n-1}^{(i)})K_n(X_{n-1}^{(i)}, X_n^{(i)})}$
13:     $\int \varphi(x_n)\pi_n(x_n)\mathrm{d}x_n \approx \frac{\sum_{i=1}^N \widetilde{W}_n(X_{n-1}^{(i)}, X_n^{(i)})\varphi_n(X_n^{(i)})}{\sum_{i=1}^N \widetilde{W}_n(X_{n-1}^{(i)}, X_n^{(i)})}$
14:     $W_n(X_{n-1}^{(i)}, X_n^{(i)}) \propto V_n(\tau^{(i)}_{n,k_n^{(i)}}, \theta^{(i)}_{n,k_n^{(i)}}, y_{n+1})\widetilde{W}_n(X_{n-1}^{(i)}, X_n^{(i)})$
15: end for
16: goto 8
```

method could be established using the results and techniques of Chopin (2004), Del Moral et al. (2006) and Johansen and Doucet (2007).

4 Simulation Study

In the first example we illustrate a minimal SMC implementation showing that substantial performance improvements can be obtained with a very slight increase in complexity. The second example shows that, using a mixture of moves and the proposed auxiliary technique, good results can be obtained without introducing additional MCMC moves.

4.1 Shot Noise Cox Process

The signal process is the intensity of a Shot noise Cox Process and takes values in $\Xi = \mathbb{R}^+$. The model is specified by the following distributions:

$$\begin{aligned} f(\tau_j|\tau_{j-1}) =& \lambda_\tau \exp(-\lambda_\tau(\tau_j - \tau_{j-1})) \times \mathbb{I}_{[\tau_{j-1},\infty)}(\tau_j), \\ q(\zeta_0) =& \exp(-\lambda_\theta \zeta_0) \times \mathbb{I}_{[0,\infty)}(\zeta_0), \\ q(\theta_j|\theta_{j-1}, \tau_j, \tau_{j-1}) =& \lambda_\theta \exp(-\lambda_\theta(\theta_j - \zeta_{\tau_j}^-)) \times \mathbb{I}_{[\zeta_{\tau_j}^-,\infty)}(\theta_j), \end{aligned}$$

where $\zeta_{\tau_j}^- = \theta_{j-1} \exp(-\kappa(\tau_j - \tau_{j-1}))$, and $F(t, \tau, \theta) = \theta \exp(-\kappa(t - \tau))$.

Given $\zeta_{(t_{n-1},t_n]}$, the observation Y_n is an inhomogeneous Poisson process with intensity $\zeta_{(t_{n-1},t_n]}$. The likelihood function is given by:

$$g(y_n|\zeta_{(t_{n-1},t_n]}) = \exp\left(-\int_{t_{n-1}}^{t_n} \zeta_s \mathrm{d}s\right) \prod_i \zeta_{y_{n,i}}$$

where $y_{n,i}$ is the time of the ith event observed in $(t_{n-1}, t_n]$. In reinsurance pricing applications, the observed events model the claims on an insurance portfolio. The jumps of the signal process model 'primary events': incidents which result in insurance claims. An approximation to the optimal filter for this model was derived in Dassios and Jang (2005). Their approximation of the optimal filter is poor when the rate of the primary event process, λ_τ is low, as it is in economically-important settings. The method we propose suffers from no such restrictions. Similar models find applications in the pricing of options, for example Centanni and Minozzo (2006).

We compare the performance of the basic VRPF approach (referred to as Scheme 1) to the proposed algorithm (Scheme 2), on data simulated from the model with the following parameter settings: $\kappa = 0.01$, $\lambda_\tau = 1/40$, $\lambda_\theta = 2/3$, $t_n - t_{n-1} = \Delta = 50$ and 500 particles. The true hidden trajectory and a histogram of the observed points are given in figure 1.

To demonstrate a simple approach to design of the proposal kernel, we used a single move; the proposal kernel is a birth move, proposing a new jump time in the interval $(t_{n-1}, t_n]$ and then drawing the associated parameter from its full conditional distribution:

$$\begin{aligned} K_n(x_{n-1}, dx_n) = {} & \delta_{k_{n-1}+1}(k_n)\delta_{\tau_{n-1,1:k_{n-1}}}(d\tau_{n,1:k_n-1}) \\ & \times \delta_{\theta_{n-1,1:k_{n-1}}}(d\theta_{n,1:k_n-1})h_n(d\tau_{n,k_n})\pi_n(d\theta_{n,k_n}|x_n \setminus \theta_{n,k_n}), \end{aligned}$$

where $h_n(d\tau_{n,k_n})$ is the proposal distribution for the time of the new jump time. This was built from a histogram of the data observed on $(t_{n-1}, t_n]$ in the following manner. Consider partitioning the interval $(t_{n-1}, t_n]$ into m bins of equal length. Let c_p be the number of observed events in the pth bin. For $p \in \{1, 2, ..., m-1\}$ define $d_p = c_{p+1} - c_p$. Then define a piecewise constant probability density function on $(t_{n-1}, t_n]$ by partitioning the interval into $m-1$ sub-intervals of equal length. In the pth such sub-interval the density has the value:

$$h_n(\tau) = \frac{\exp(d_p)}{\sum_{q=1}^{m-1} \exp(d_q)}, \qquad \tau \in (t_{n-1} + (p-1)\delta, t_{n-1} + p\delta],$$

where $\delta = (t_n - t_{n-1})/(m-1)$. The corresponding component of the backwards kernel is:

$$L_{n-1}(x_n, dx_{n-1}) = \delta_{k_n-1}(k_{n-1})\delta_{\tau_{n,1:k_n-1}}(d\tau_{n-1,1:k_{n-1}})\delta_{\theta_{n,1:k_n-1}}(d\theta_{n-1,1:k_{n-1}}).$$

Systematic resampling was applied when the ESS dropped below 40%. After resampling, a trans-dimensional Metropolis-Hastings kernel consisting of a sequence of three moves was applied. Firstly, a birth/death reversible jump move, with the birth being made by drawing a new jump time from the uniform distribution on $(\tau_{n,k_n}, t_n]$ and then drawing the parameter from the full conditional distribution. Secondly, a perturbation of the most recent jump time, τ_{n,k_n}, drawing from a Gaussian kernel centred at τ_{n,k_n} and truncated to the interval $(\tau_{n,k_n-1}, t_n]$. Thirdly, a perturbation of the parameter associated with the most recent jump time, drawing from the full conditional distribution.

Resampling was required during fewer than 40% of the algorithm's iterations. Despite the fact that the proposal kernel consisted only of a birth move, it was found that the algorithm did not significantly over-fit the data: for the date shown in figure 3a), the true number of jumps is 34 and for a typical run, the MAP number of jumps was 39. The MMSE filtering estimate of ζ_t obtained using the proposed algorithm is shown in figure 1.

Figure 2 shows a count of the number of unique particles, over time, at the final iteration of the algorithm. These plots were constructed as follows. The particle histories $\{\zeta_{[0,2000]}^{(i)}\}_{i=1}^{N}$ were stored at the final iteration of the algorithm. For each value of t, the number of unique particles in $\{\zeta_t^{(i)}\}_{i=1}^{N}$ were then counted and plotted against t.

The diversity of particle locations alone does not tell us everything about the efficiency of the algorithm. Whilst there may be many unique particles, the importance weights may have high variance, in which case the quality of the particle approximation will be low. In order to portray this characteristic, we also plot the same quantities, but after resampling is applied (whilst resampling can increase the variance of an estimate made from the particle set, for systematic resampling this increase is typically extremely small). These quantities demonstrate the degree of degeneracy of both the particle locations and the degeneracy of the importance weights. Pre-resampling, Scheme 1 exhibits some diversity in the recent history of particle locations, note that the number of unique particles is less than the total number. This is an example of the phenomenon described in section 2.3. Degeneracy of the importance weights means that this diversity is reduced when resampling occurs. Scheme 2 exhibits diversity in particle locations much further back in time, with all particles being unique in the recent history. Reduction in the number of unique particles when resampling occurs is less severe than for Scheme 1, demonstrating that the importance weights have lower variance.

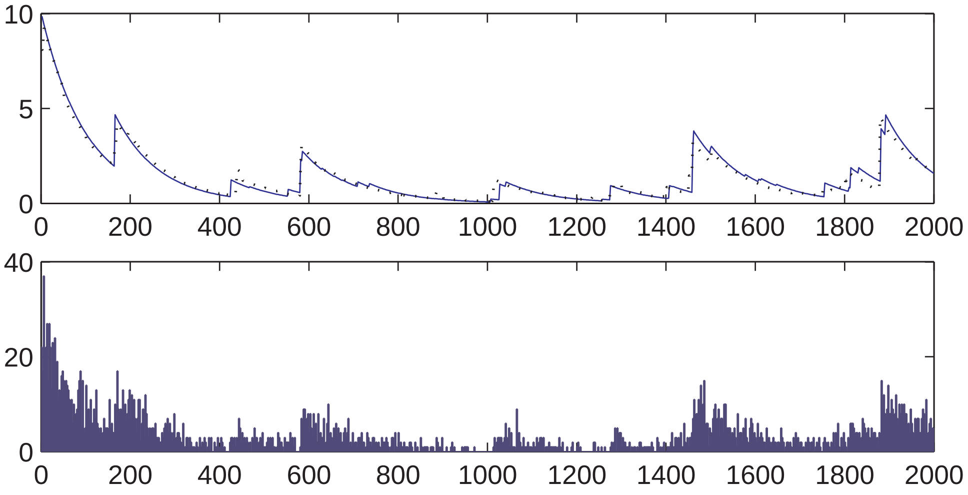


Figure 1: SNCP model. Top: True intensity (solid) and MMSE filtering estimate (dotted) against time (secs.). Bottom: histogram of realised observations with bin length 2.5.

4.2 Object Tracking

This example illustrates the performance of the proposed methods on a constant acceleration tracking model applied to the benchmark fighter aircraft trajectory depicted in figure 3a),

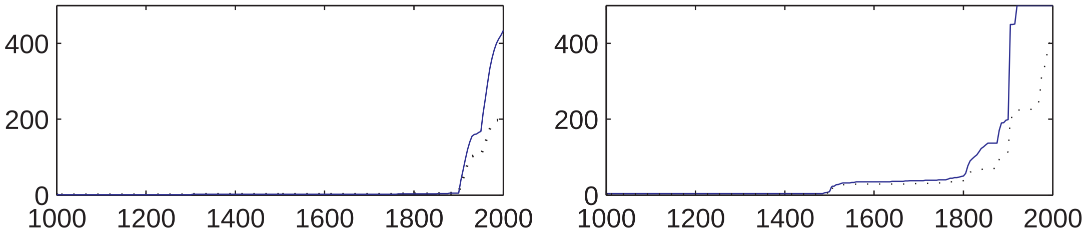


Figure 2: Unique particles at the final iteration vs time (secs.). Solid: Pre-resampling. Dotted: Post-resampling. Left: Scheme 1. Right: Scheme 2.

which has duration 185 seconds. The model is as specified in subsection 2.2, with acceleration components being drawn independently from an isotropic Gaussian distribution of mean zero and standard deviation $\sigma_\theta = 10\text{m/s}^2$. 37 additive, isotropic Gaussian observations of the position of the aircraft, with standard deviation $\sigma_y = 200\text{m}$, were generated at intervals of $t_n - t_{n-1} = \Delta = 5\text{s}$, also shown in figure 3a). The inter-jump times were chosen to be Gamma distributed, with shape and scale parameters 10 and 2.5 respectively, corresponding to a mean inter-arrival time 25s. For this conditionally linear-Gaussian model it is possible to analytically integrate out the parameters $\theta_{0:k_n}$, and only jump times need be sampled. This is the direct analogue of Rao-Blackwellised SMC for discrete-time filtering, see Doucet et al. (2001) and Chopin (2004) for a discussion. Comparisons were made between three schemes described below. In all cases, systematic resampling was employed.

Scheme 1. The basic VRPF algorithm, sampling from the prior, see the appendix for specification of the kernels. Resampling was performed when the ESS fell below 50%.

Scheme 2. A forward mixture kernel with two components. The first component is a birth move which adds a single point uniformally in $(\tau_{n,k_n-1}, t_n]$:

$$K_{n,1}(x_{n-1}, dx_n) = \delta_{k_{n-1}+1}(k_n)\delta_{\tau_{n-1,1:k_{n-1}}}(d\tau_{n,1:k_n-1}) \times \frac{d\tau_{n,k_n}}{t_n - \tau_{n,k_n-1}}\mathbb{I}_{(\tau_{n,k_n-1},t_n]}(\tau_{n,k_n}).$$

For this move, the corresponding component of the backwards kernel is:

$$L_{n-1,1}(x_n, dx_{n-1}) = \delta_{k_n-1}(k_{n-1})\delta_{\tau_{n,1:k_n-1}}(d\tau_{n-1,1:k_{n-1}}).$$

The second component is an adjustment move in which a Gaussian random walk kernel, restricted to $(\tau_{n,k_n-1}, t_n]$ is applied to the most recent jump time. This component of the kernel is given by:

$$\begin{aligned} K_{n,2}(x_{n-1}, dx_n) \propto\ & \mathbb{I}_{(\tau_{n-1,k_{n-1}-1},t_n]}(\tau_{n,k_n})\mathcal{N}(\tau_{n-1,k_{n-1}}, \sigma_a^2)d\tau_{n,k_n} \\ & \times \delta_{k_{n-1}}(k_n)\delta_{\tau_{n-1,1:k_{n-1}-1}}(d\tau_{n,1:k_n-1}). \end{aligned}$$

A Gaussian kernel of the same variance, restricted to $(\tau_{n-1,k_{n-1}-1}, t_{n-1}]$, is used for the corresponding component of the backward kernel:

$$L_{n-1,2}(x_n, dx_{n-1}) \propto \mathbb{I}_{(\tau_{n-1,k_{n-1}-1},t_{n-1}]}(\tau_{n-1,k_{n-1}})\mathcal{N}(\tau_{n,k_n}, \sigma_a^2)d\tau_{n-1,k_{n-1}}$$
$$\times\, \delta_{k_n}(k_{n-1})\delta_{\tau_{n,1:k_n-1}}(d\tau_{n-1,1:k_{n-1}-1}).$$

The forward mixture weights were defined as in (14), with the backward mixture weights set to uniform. Resampling was performed when the ESS fell below 50%.

Scheme 3. As 2. but with an auxiliary scheme using an approximation of the following auxiliary weighting function:

$$\begin{aligned} V_n(\tau_{n,k_n}, \theta_{n,k_n}, y_{n+1}) = &\frac{S(t_{n+1}, \tau_{n,k_n})}{S(t_n, \tau_{n,k_n})} p(y_{n+1}|y_{1:n}, k_{n+1} = k_n, \tau_{n,1:k_n}) \\ &+ \frac{1}{(S(t_n, \tau_{n,k_n}))^2} \left\{ \int_{(t_n,t_{n+1}]} f(\tau_{n+1,k_{n+1}}|\tau_{n,k_n}) S(t_n, \tau_{n+1,k_{n+1}}) \right. \\ &\left. \times\, p(y_{n+1}|y_{1:n}, k_{n+1} = k_n + 1, \tau_{n+1,1:k_{n+1}}) \mathrm{d}\tau_{n+1,k_{n+1}} \right\}. \end{aligned}$$

The one-dimensional integral is approximated using standard numerical techniques (note this does not affect the theoretical validity of the algorithm). The motivation for this auxiliary weighting function is that it approximates the predictive likelihood. Resampling was applied at every iteration.

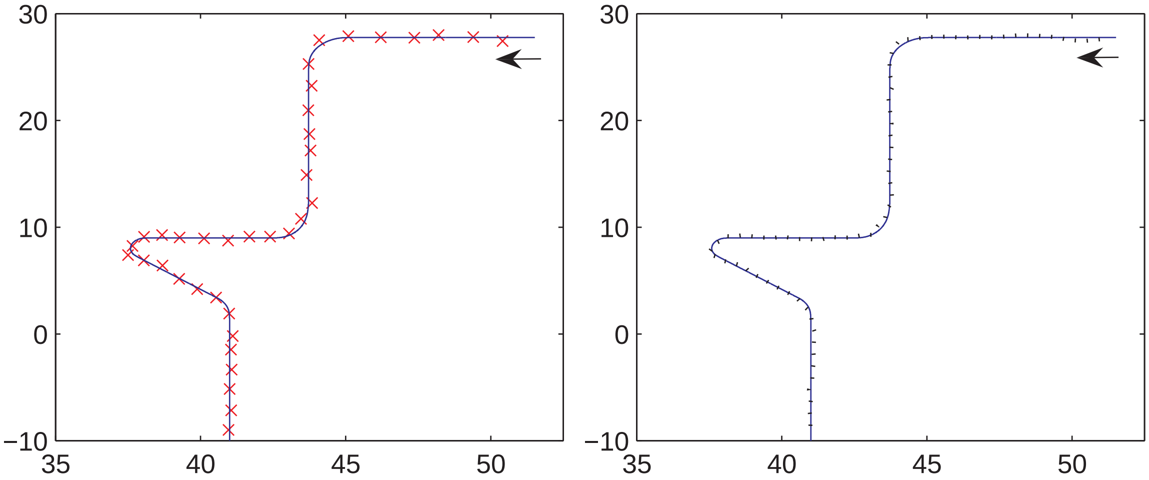


Figure 3: Benchmark 2D position trajectory (solid). The trajectory begins around (66,29), with the aircraft executing several manoeuvres before the trajectory terminates after a duration of 185 seconds. Left: observations with additive Gaussian noise. Right: MMSE filtering estimate (dotted). Scale is in km.

It was found that all three schemes yielded roughly the same error performance in estimating ζ_{t_n} from each π_n and little decrease in estimation error was observed using more than 200 particles. This can be attributed to the Rao-Blackwellised nature of the model. For an example of a model in which the parameters are not integrated out, see Whiteley et al. (2007); in this instance the VRPF exhibits significantly greater error in this filtering task, and is substantially outperformed by an algorithm which includes adjustment moves. MMSE estimates of the position of the vehicle using Scheme 3, and made before resampling, are shown in figure 3b). However, as more of the recent history of the trajectory is considered, including the times of the jumps in acceleration, the superior performance of the proposed method becomes apparent. Figure 4 provides insight into the relationship between the particle diversity and weight degeneracy for the three algorithms, at the final iteration of the algorithm.

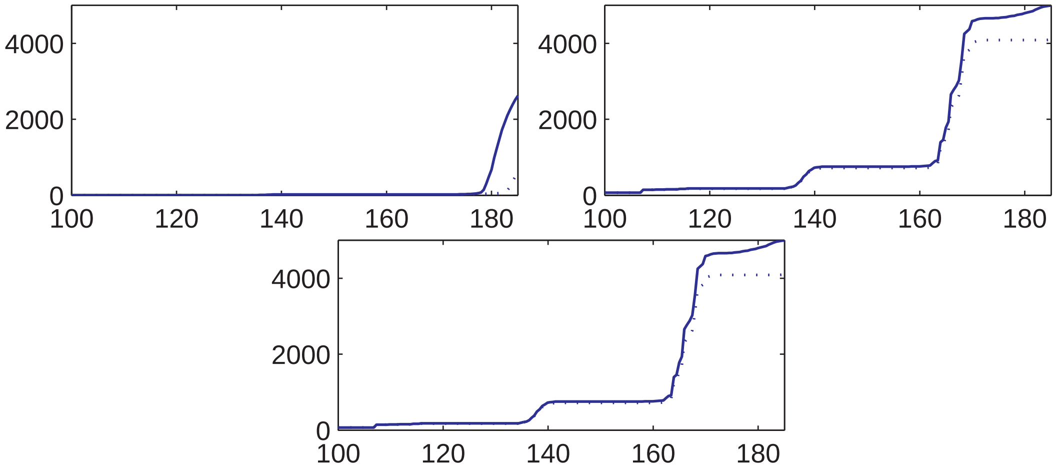
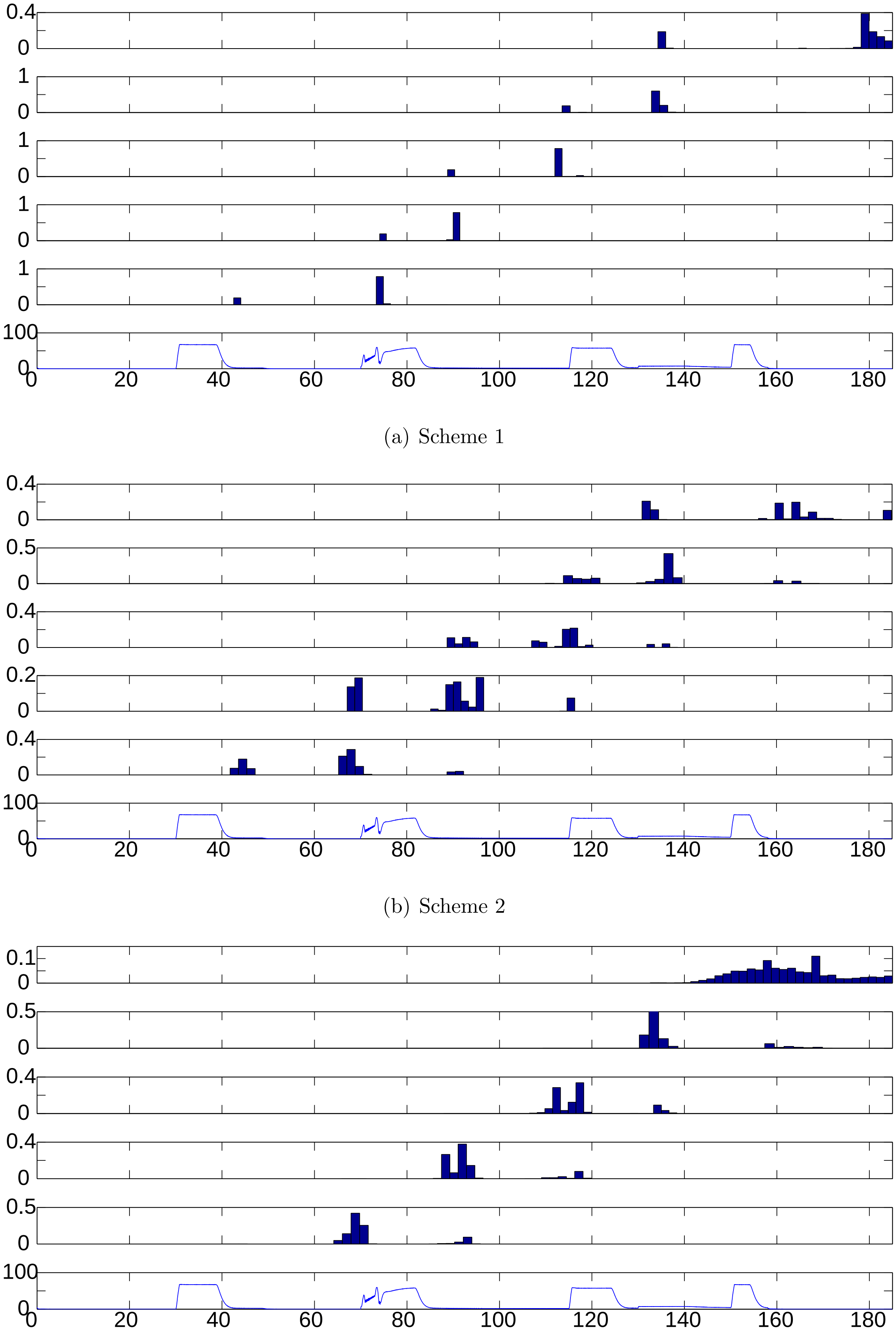


Figure 4: Unique particles at the final iteration vs time (secs). Pre-resampling (solid), Post-resampling (dotted). Top left: Scheme 1. Top Right: Scheme 2. Bottom: Scheme 3.

In order to show the diversity of particle set, we plot the number unique particles at the final iteration, as a function of time. Again we consider the particle set before and after resampling, but note that in the case of the Scheme 3, for the purposes of these diagnostics, we applied resampling at the final iteration according to the importance weights from which an estimate over the target distribution would be drawn. The difference between the pre and post-resampling plots for each algorithm indicate the variance of the importance weights: when the variance is high, the number of unique particles is relatively low post-resampling. For these tests the adjustment kernel standard deviation was set to $\sigma_a = \Delta/1000$s.

From figure 4 it can be seen that Scheme 1 exhibits particle diversity only in the very recent history, even before resampling. Scheme 2 exhibits much more diversity, especially between the times of the last and penultimate observations. However, much of this diversity is lost upon resampling. The plot of Scheme 3 shows that a significant proportion of diversity arising from the adjustment move and the births in $(\tau_{n,k-1}, t_n]$ is not lost after resampling. It



(a) Scheme 1

(b) Scheme 2

(c) Scheme 3

Figure 5: Weighted histograms of particle jump times at final iteration, pre-resampling against time (secs.). In each sub figure, top to second bottom: $\tau_{n,k}$, $\tau_{n,k-1}$, $\tau_{n,k-2}$, $\tau_{n,k-3}$, $\tau_{n,k-4}$ and bottom: true acceleration magnitude (ms^{-2}). $\sigma_a = \Delta/1000$. $N = 5000$.

was found that choosing σ_a to be of order less than that of the inter-observation time avoided weight degeneracy. Figure 5 compares the performance of algorithms in terms of estimating jump locations. The use of the adjustment move in Scheme 2 yields a small improvement over Scheme 1. However, the results for Scheme 3 show a marked improvement. Note that, in the model, jumps occur in the acceleration but it is only the position of the vehicle which is observed. Therefore optimal estimates of jump times in the very recent history should exhibit large variance, the results for Scheme 3 are commensurate with this.

5 Conclusions

We have presented a broad class of algorithms for the filtering of Piecewise Deterministic processes and demonstrated that this class incorporates existing algorithms as particular cases. The significance of this work is that the proposed algorithms allow for the use of principled techniques which lead to substantial improvements in efficiency when compared with existing techniques.

6 Supplemental Materials

Appendix: Specification of VRPF algorithm. (Appendix.pdf, portable document format)

Matlab Code: Various functions for implementing the algorithms. (smc.m, Matlab m-file)

References

Blair, W., Watson, G., Kirubarajan, T., and Bar-Shalom, Y. (1998), "Benchmark for radar allocation and tracking in ECM," *IEEE Transactions on Aerospace and Electronic Systems*, 34, 1097–1114.

Cappé, O., Godsill, S., and Moulines, E. (2007), "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," *Proceedings of the IEEE*, 96, 899 –924.

Centanni, S. and Minozzo, M. (2006), "Estimation and filtering by reversible jump MCMC for a doubly stochastic Poisson model for ultra-high-frequency financial data," *Statistical Modelling*, 6, 97–118.

Chopin, N. (2004), "Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference," *Annals of Statistics*, 32, 2385–2411.

— (2006), "Dynamic detection of change points in long time series," *Annals of the Institute of Statistical Mathematics*, 59, 349–366.

Dassios, A. and Jang, J. (2005), "Kalman-Bucy filtering for linear system driven by the Cox process with shot noise intensity and its application to the pricing of reinsurance contracts," *Journal of Applied Probability*, 42, 93–107.

Davis, M. (1984), "Piecewise Deterministic Markov Processes: a general class of non-diffusion stochastic models," *Journal Royal Statistical Society B*, 46, 353–388.

Del Moral, P., Doucet, A., and Jasra, A. (2006), "Sequential Monte Carlo Samplers," *Journal of the Royal Statistical Society B*, 63, 411–436.

Doucet, A., Gordon, N., and Krishnamurthy, V. (2001), "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, 49, 613–624.

Doucet, A. and Johansen, A. M. (2009), "A Tutorial on Particle filtering and smoothing: Fifteen years later," in *Oxford Handbook of Nonlinear Filtering*, eds. Crisan, D. and Rozovsky, B., Oxford University Press, to appear.

Doucet, A., Montesano, L., and Jasra, A. (2006), "Optimal Filtering For Partially Observed Point Processes Using Trans-Dimensional Sequential Monte Carlo," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processsing, (ICASSP).*

Gilks, W. and Berzuini, C. (2001), "Following a moving target - Monte Carlo inference for dynamic Bayesian Models," *Journal of the Royal Statistical Society B*, 63, 127–146.

Godsill, S. (2007), "Particle filters for continuous-time jump models in tracking applications," *ESAIM: Proceedings*, 19, 39–52.

Godsill, S., Vermaak, J., Ng, K.-F., and Li, J.-F. (2007), "Models and Algorithms for Tracking of Manoeuvring Objects using Variable Rate Particle Filters," *Proceedings of IEEE*, 95, 925–952.

Godsill, S. J. and Vermaak, J. (2004), "Models and Algorithms for tracking using trans-dimensional sequential Monte Carlo," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, pp. 976–979.

Jacobsen, M. (2006), *Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes*, Probability and its Applications, Boston: Birkhäuser.

Johansen, A. M. and Doucet, A. (2007), "Auxiliary Variable Sequential Monte Carlo Methods," Tech. Rep. 07:09, University of Bristol, Department of Mathematics – Statistics Group, University Walk, Bristol, BS8 1TW, UK.

— (2008), "A Note on the Auxiliary Particle Filter," *Statistics and Probability Letters*, 78, 1498–1504.

Kong, A., Liu, J. S., and Wong, W. H. (1994), "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, 89, 278–288.

Lasdas, V. and Davis, M. (1989), "A piecewise deterministic process approach to target motion analysis [sonar]," in *Proceedings of the 28th IEEE Conference on Decision and Control, Tampa, FL, USA*, vol. 2, pp. 1395–1396.

Maskell, S. (2004), "Joint Tracking of Manoeuvring Targets and Classification of their manoeuvrability," *EURASIP Journal on Applied Signal Processing*, 15, 2339–2350.

Pitt, M. K. and Shephard, N. (1999), "Filtering via Simulation: Auxiliary Particle Filters," *Journal of the American Statistical Association*, 94, 590–599.

Sworder, D. D., Kent, M., Vojak, R., and Hutchins, R. G. (1995), "Renewal Models for Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, 31, 138–150.

Whiteley, N., Johansen, A. M., and Godsill, S. (2007), "Efficient Monte Carlo filtering for discretely observed jumping processes," in *Proceedings of IEEE Statistical Signal Processing Workshop, Madison, Wisconsin, USA*, pp. 89–93.