EFFICIENT MONTE CARLO FILTERING FOR DISCRETELY OBSERVED JUMPING PROCESSES

Nick Whiteley[1], *Adam M. Johansen*[2] *and Simon Godsill*[1]

1 - University of Cambridge, Department of Engineering, Trumpington Street, Cambridge, CB2 1PZ, UK

2 - University of Bristol, Department of Mathematics, University Walk, Bristol, BS8 1TW, UK

ABSTRACT

This paper addresses a tracking problem in which the unobserved process is characterised by a collection of random jump times and associated random parameters. We construct a scheme for obtaining particle approximations to the posterior distributions of interest in the framework of sequential Monte Carlo (SMC) samplers [1]. We describe efficient sampling schemes and demonstrate that two existing schemes can be interpreted as particular cases of the proposed method. Results are provided which illustrate the performance improvements possible with our approach.

Index Terms— Monte Carlo methods, Nonlinear filters, Continuous time systems

1. INTRODUCTION

Within the Bayesian paradigm, the task of optimal filtering corresponds to obtaining, recursively in time, the posterior distribution of an unobserved stochastic process, given noisy observations made over time. The filtering model has many applications in signal processing, not least in *tracking* where the hidden process models the evolution of a manoeuvring object. The aim is then to estimate the trajectory of some object, given observations made by some noisy sensor.

Tracking is commonly cast as a discrete time filtering problem in which the hidden process is Markov and the observations are conditionally independent, given the state of the system. In many cases of interest the state-space model is non-linear and non-Gaussian, and exact inference is intractable. Approximation methods must, therefore, be employed. SMC methods, [2], approximate the sequence of posterior distributions by a collection of weighted samples, termed *particles*.

It has been demonstrated that, in some situations, the trajectory of a manoeuvring target may be more parsimoniously modelled by a possibly non-Markovian, continuous time process [3]. If such models are to be employed in practice, accurate and computationally efficient inference schemes need be developed. The contribution of this paper is the development of such schemes. Specifically, we address the filtering of a broad class of semi-Markov processes, employing sequential Monte Carlo techniques to approximate the distributions of interest.

1.1. Problem Statement

We first define the *signal process* $(\zeta_t)_{t\geq 0}$, where each ζ_t takes a value in a state space Ξ (e.g., n-dimensional Euclidean space), and a sequence of noisy *observations* $(Y_n)_{n\in\mathbb{N}}$, where each $y_n \in \mathbb{R}^{d_y}$ with $d_y \in \mathbb{N}$, which are independent of one another and, conditional upon the signal process at the observation time, of the remainder of the signal process. Our aim is to obtain iteratively, at each observation time, the conditional distribution of the signal process given the collection of observations up to that time.

1.2. Model Specification

We begin with a formal specification of the model, before providing an intuitive explanation; a simple example is provided in section 1.3.

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

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

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

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

The right-continuous 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 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 is deterministic and subject to the condition that $F(\tau_j, \tau_j, \theta_j) = \theta_j$, $\forall j \in \mathbb{N}$.

It is easy to interpret this dynamic model: a realisation of the signal process evolves from the initial condition ζ_0 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 according to F until τ_2, at which time the signal acquires the new value θ_2, and so on.

The nth observation of the signal process, Y_n, is made at time point t_n via some function H, in the presence of an independent noise component V_n:

$$Y_n = H(\zeta_{t_n}, V_n).$$

The distribution of V_n, together with H, induces a likelihood function $g(y_n|\zeta_{t_n})$.

We will be especially interested in the number of jumps occurring in each 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:

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

where $S(t, \tau)$ is the survivor function associated with the transition density $f(\tau_j|\tau_{j-1})$:

$$S(t,\tau) = 1 - \int_{\tau}^{t} f(s|\tau)\mathrm{d}s.$$

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 $\theta_{1:k_n}$. We define a sequence $(X_n)_{n\in\mathbb{N}}$, where, omitting explicit n-indexing of all components for brevity, $X_n = (k_n, \zeta_0, \theta_{1:k_n}, \tau_{1:k_n})$ takes its values in the disjoint union:

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

with $\mathbb{R}^k \supset \Upsilon_{n,k} = \{\tau_{1:k} : 0 < \tau_1 < \cdots < \tau_k \leq t_n\}$.

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 distribution of X_n, because, by construction, the signal process is a deterministic function of the jump times and parameters. This posterior distribution, up to a constant of proportionality, has the form:

$$\pi_n(x_n) \propto p_n(k_n, \tau_{1:k_n}) \times q_0(\zeta_0) \prod_{j=1}^{k_n} q(\theta_j|\theta_{j-1}, \tau_j, \tau_{j-1}) \prod_{p=1}^{n} g(y_p|\zeta_{t_p}). \quad (1)$$

The marginal distribution $\pi_n(\tau_{k_n}, \theta_{k_n})$ provides sufficient information to obtain the filtering distribution, $p(\zeta_{t_n}|y_{1:n})$. Although, in the following, we consider only filtering distributions of the form $p(\zeta_{t_n}|y_{1:n})$, i.e. for the signal at the times of the observations, just as in the standard discrete time filtering scenario, the proposed method can be straightforwardly modified to deal with other filtering and smoothing distributions. Exact inference for this model is intractable and in section 2 we describe Monte Carlo schemes for obtaining sample-based approximations to posterior distributions.

We note that obtaining these distributions amounts to solution of the optimal Bayesian filtering problem for the model described above. Obtaining particle approximations of these distributions provides us with a computationally tractable method for obtaining arbitrarily good approximations of these distributions. Determining whether this is a good description of a particular physical system is a modelling problem which we do not consider here.

1.3. A Motivating Example

A vehicle manoeuvres according to standard, piece-wise constant acceleration dynamics. Each parameter may be decomposed into x and y components, each containing a position, velocity and acceleration value, for brevity we write,

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

At time zero the vehicle has position, velocity and acceleration ζ_0. At time τ_1, the acceleration of the vehicle jumps to a new, random value according to $q(\theta_j|\theta_{j-1}, \tau_{j-1}, \tau_j)$ etc. 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,

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

The component of F in the y-direction is equivalent. This model is considered suitable for the benchmark fighter aircraft trajectory from [4], shown in figure 1.

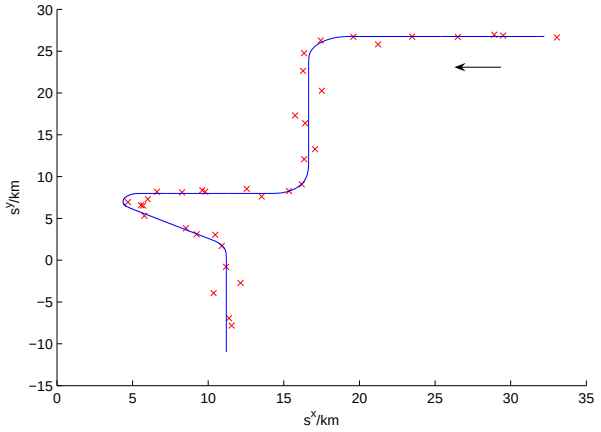

Fig. 1. Benchmark 2D position trajectory (solid) and additive Gaussian observations (crosses).

2. METHODOLOGY

2.1. Previous Approaches

Inference schemes based on the ideas of sequential importance sampling and resampling, upon which the particle filter is built, have been devised for the process of interest. Whilst it is possible, in some circumstances to consider discrete time approximations to the process of interest the nature of this approximation is not always clear and the error which it introduces is not easy to control. Consequently, we will consider only techniques in which no modelling approximations are employed.

The variable rate particle filter (VRPF) of [5, 3] is one such scheme, which samples a random sequence of jump times on the interval $[0, t_n]$ by drawing recursively from $f(\tau_j|\tau_{j-1})$, until a stopping time criterion is met. One could equivalently sample first from $p_n(k_n)$ and then from $f_n(\tau_{1:k}|k_n)$. A similar method was presented independently in [6]. Note that the proposed approach permits a range of more effective proposal moves.

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) these schemes can result in the propagation of multiple copies of the same particle. More computationally efficient methods dealing with this issue were proposed in [3], but there remains a disadvantage in terms of the variance of state estimates.

2.2. SMC Samplers

The SMC samplers framework of [1] is a very general method for obtaining a set of samples from a sequence of distributions which exist on the same or different spaces. This can be viewed as a generalisation of the standard SMC method in which the target distribution exists on a space of strictly increasing dimension. The use of these techniques for trans-dimensional inference was further discussed in [7]; SMC samplers have recently been applied to similar trans-dimensional problems in the context of point processes [8].

It is not possible to give a thorough exposition of the SMC samplers approach here, but we will try to include sufficient detail for our purposes. Given a sequence of distributions $(\pi_n)_{n\in\mathbb{N}}$ on a sequence of spaces $(E_n)_{n\in\mathbb{N}}$ from which we wish to obtain sets of weighted samples, we construct a sequence of distributions on a sequence of

spaces of increasing dimension which admit the distributions of interest as marginals, by defining:

$$\widetilde{\pi}_n(x_{1:n}) = \pi_n(x_n) \prod_{p=n-1}^{1} L_n(x_{n+1}, x_n),$$

where L_n is a Markov kernel from space E_{n+1} to E_n. Standard SMC methods can now be applied on this space, by propagating samples forward from one distribution to the next according to a sequence of Markov kernels, $(K_n)_{n\geq 2}$, and correcting for the discrepancy between the proposal and the target distribution by incremental importance weights of the form:

$$w_n(x_{n-1}, x_n) \propto \frac{\pi_n(x_n)L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}.$$

It is important to ensure that a significant fraction of the particle set have non-negligible weights. The effective sample size (ESS), introduced by [9], is an approximation of a quantity which describes the effective number of iid samples to which the set corresponds.

Denoting by $\left\{W^{(i)}\right\}$ the normalized weights, the ESS is defined as $ESS = \left[\sum_{i=1}^{N} W^{(i)^2}\right]^{-1}$. Resampling should be carried out after any iteration which causes the ESS 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.

It can be shown (again, see [1]) that the optimal form for the Markov kernels L_n – in the sense of minimising the variance of the importance weights if resampling occurs at every time step – is given by:

$$L_n^{opt}(x_{n+1}, x_n) = \frac{\pi_n(x_n)K_{n+1}(x_n, x_{n+1})}{\int \pi_n(x)K_{n+1}(x, x_{n+1})dx}. \qquad (2)$$

In practice it is important to choose a sequence of kernels which are as close to the optimal case as possible to prevent the variance of the importance weights from becoming extremely large.

The proposal kernel K_n can be chosen to be a mixture of different move types:

$$K_{n+1}(x_n, x_{n+1}) = \sum_{m=1}^{M} \alpha_m K_{n+1,m}(x_n, x_{n+1}), \quad \sum_{m=1}^{m} \alpha_m = 1,$$

and in this case it follows from (2) that the optimal backward kernel can also be expressed as a mixture.

2.3. Trans-Dimensional SMC Filtering

By applying the SMC samplers method to the sequence of distributions $(\pi_n(x_n))_{n\in\mathbb{N}}$, see (1), we obtain a recursive scheme which propagates a particle approximation to each marginal distribution $\pi_n(\tau_{k_n}, \theta_{k_n})$, and thus to $p(\zeta_{t_n}|y_{1:n})$.

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 described below. This allows more accurate state estimation for the same computational cost. At the n^{th} iteration, the algorithm yields a set of N particles, $\{(k_n\ \tau_{k_n}\ \theta_{k_n})^{(i)}\ ,\ w_n^{(i)}\}_{i=1}^{N}$. This approximates the filtering distribution for the signal process via:

$$P(\zeta_{t_n} \in d\zeta|y_{1:n}) = \sum_{i=1}^{N} w_n^{(i)} \delta_{\zeta_{t_n}^{(i)}}(d\zeta),$$

where $\zeta_{t_n}^{(i)} = F(t_n, \tau_{k_n}^{(i)}, \theta_{k_n}^{(i)})$.

The proposed algorithm is described in algorithm 1. Given a particular model, all that is necessary to implement such an algorithm is a proposal distribution and an associated auxiliary kernel. The choice of these elements will be discussed in the next section and further detailed in the case of the examples provided below.

Initialisation, $n = 1$:
for $i = 1$ to N **do**
  $X_1^{(i)} \sim q_1$
  {where q_1 is some importance distribution.}
  $W_1^{(i)} \propto \frac{\pi_1(X_1^{(i)})}{q_1(X_1^{(i)})}$
end for
Iteration, $n \leftarrow n + 1$:
Resample if necessary (when the effective sample size falls below a pre-determined threshold, for example).
Sample rejuvenation can be conducted at this stage by applying a π_n-invariant Markov kernel to each particle.
for $i = 1$ to N **do**
  $X_n^{(i)} \sim K_n(X_{n-1}^{(i)}, \cdot)$
  $W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(X_n^{(i)})L_{n-1}(X_n^{(i)}, X_{n-1}^{(i)})}{\pi_{n-1}(X_{n-1}^{(i)})K_n(X_{n-1}^{(i)}, X_n^{(i)})}$
end for

Algorithm 1: A Basic Jumping Process Particle Filter

2.4. Choice of Forward Kernel

The design of the proposal kernel plays a significant rôle in the performance of the algorithm. In order to minimise the variance of the importance weights, it must be well matched to the observations. A mixture kernel is suitable for the trans-dimensional problem at hand, for example consisting of the following moves:

Birth Move. The dimensionality is incremented, $k_n = k_{n-1}+1$, a new jump, τ_{k_n} is proposed uniformally in $(\tau_{k_{n-1}}, t_n]$, and a new parameter is then drawn from the full conditional $\pi_n(\cdot|x_n \setminus \theta_{k_n})$, where $x_n \setminus \theta_{k_n}$ denotes all components of x_n other than θ_{k_n}. It can be shown that this is the conditionally optimal distribution in terms of minimising the variance of the importance weights. If $\tau_{k_n} \leq t_{n-1}$ this amounts to altering the trajectory $(\zeta_t)_{t\in[\tau_{k_n}, t_{n-1}]}$ and extending the trajectory onto $(t_{n-1}, t_n]$. In this case, and denoting by ζ'_t the new trajectory, the weight expression is:

$$w_n(x_{n-1}, x_n) = \frac{S(t_n, \tau_{k_n})f(\tau_{k_n}|\tau_{k_{n-1}})(t_n - \tau_{k_{n-1}})}{S(t_{n-1}, \tau_{k_{n-1}})} \times \frac{q(\theta_{k_n}|\theta_{k_{n-1}}, \tau_{k_n}, \tau_{k_{n-1}})}{\pi_n(\theta_{k_n}|x_{n-1} \setminus \theta_{k_{n-1}})} \times \frac{\prod_{p=r}^{n} g(y_p|\zeta'_{t_p})}{\prod_{p=r}^{n-1} g(y_p|\zeta_{t_p})},$$

where

$$r = \inf\{n : t_n \geq \tau_{k_n}\}.$$

An alternative, suboptimal choice is to propose parameters from the prior, $q(\cdot|\theta_{j-1}, \tau_{j-1}, \tau_j)$.

Update Move. The dimensionality, jump locations and parameter values are maintained. In this case,

$$w_n(x_{n-1}, x_n) = \frac{S(t_n, \tau_{k_n})g(y_n|\zeta_{t_n})}{S(t_{n-1}, \tau_{k_n})}.$$

Adjustment Move. The dimensionality is maintained and the most recent parameter, θ_{k_n}, is re-drawn from the full conditional

distribution $\pi_n(\cdot|x_n \setminus \theta_{k_n})$, yielding a new parameter value θ'_{k_n}. If $\tau_{k_n} \leq t_{n-1}$ this amounts to altering the trajectory $(\zeta_t)_{t\in(\tau_{k_n},t_{n-1}]}$ and extending the trajectory onto $(t_{n-1}, t_n]$. The weight expression is:

$$w_n(x_{n-1}, x_n) = \frac{S(t_n, \tau_{k_n})q(\theta'_{k_n}|\theta_{k_n-1}, \tau_{k_n}, \tau_{k_n-1})}{S(t_{n-1}, \tau_{k_n})q(\theta_{k_n}|\theta_{k_n-1}, \tau_{k_n}, \tau_{k_n-1})} \times \frac{\pi_{n-1}(\theta_{k_n}|x_{n-1} \setminus \theta_{k_n})}{\pi_n(\theta'_{k_n}|x_{n-1} \setminus \theta_{k_n})} \times \frac{\prod_{p=r}^{n} g(y_p|\zeta'_{t_p})}{\prod_{p=r}^{n-1} g(y_p|\zeta_{t_p})},$$

where

$$r = \inf\{n : t_n \geq \tau_{k_n}\}.$$

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.

Other Moves. It is possible to construct a variety of other moves which alter the recent history of each particle. For example, after resampling, a π_n-invariant Metropolis-Hastings kernel can be used to perturb the position of the most recent jump or add/remove jumps. Such moves are important if fixed-lag smoothing is to be performed.

Kernel Mixture Weights. A technical requirement of importance sampling schemes is that support of the proposal distribution includes that of the posterior distribution. Therefore a forward kernel capable of proposing any positive number of births in the interval $(t_{n-1}, t_n]$ must be employed. However, the mixture weight associated with this component may be made small when the transition density $f(\cdot|\tau_{j-1})$ assigns very little mass to short inter-arrival times.

For any given combination of move types, the forward kernel mixture weights play a significant role in the importance weights. For a kernel consisting of moves which each propose a different increment to the number of jumps, the kernel components have disjoint support. In this case the forward mixture weight corresponding to the move executed should simply be multiplied into the denominator of the importance weight.

By employing a mixture of update moves and prior birth moves on the interval $(t_{n-1}, t_n]$, in proportions specified by the prior on the dimensionality parameter k_n, we obtain algorithms as in [6] and [5].

It should not be noted that the above moves are such that, at each iteration, the algorithm requires storage of only a fixed-length history for each particle. Simple restrictions can be imposed if there is a need to store only a fixed number of observations.

3. RESULTS

We present results obtained by applying this algorithm and the standard VRPF (i.e. proposals from the prior) to a simple two dimensional tracking model and a more complicated model which is difficult to deal with efficiently using conventional methods.

3.1. Example 1

We consider the model as described in section 1.3 in the case that $f(\cdot|\tau_{j-1})$ is exponential with mean 5Δ and acceleration parameters are i.i.d. zero mean Gaussian with standard deviation $0.05\ m/s^2$. Along the position trajectory, $n = 37$ additive, zero mean, isotropic Gaussian noise observations were generated with time interval $\Delta =$ 5s and standard deviation $\sigma_y = 500$m. Examples are shown in figure 1. The forward kernel was chosen to have equal proportions of birth and adjustment moves using the true conditional distribution as the proposal combined with the optimal backward kernels. Systematic resampling was applied when the ESS dropped below 50%.

A root mean square error (RMSE) criterion was used to assess the performance of the proposed algorithm compared to the VRPF, based on filtering estimates of the vehicle position at the times of each observation, over $M = 200$ observation realizations.

$$RMSE = \frac{1}{n}\sum_{p=1}^{n}\left[\frac{1}{M}\sum_{m=1}^{M}(s^x_{t_p,m} - \hat{s}^x_{t_p,m})^2 + (s^y_{t_p,m} - \hat{s}^y_{t_p,m})^2\right]^{1/2}$$

where $s^x_{t_p,m}$ and $\hat{s}^x_{t_p,m}$ are respectively the true and MMSE estimated position in the x direction at the time of the pth observation of the mth run.

The results in table 1 indicate the proposed method out-performs the VRPF, especially when the number of particles is small. The computational cost of the VRPF is similiar to that of the TDSMC algorithm for this model. This is due to the fact short inter-arrival times occur frequently under the prior so the VRPF generates more random numbers, but the TDSMC algorithm requires more resources to calculate kernel parameters and importance weights.

	VRPF		**TDSMC**	
N	**RMSE / km**	**CPU / s**	**RMSE / km**	**CPU / s**
50	10.24	0.97	0.61	0.76
100	4.89	1.93	0.58	1.52
250	1.76	4.90	0.56	3.90
500	0.85	10.01	0.55	7.75
1000	0.74	20.12	0.54	15.62
2500	0.65	51.74	0.53	38.82
5000	0.63	104.91	0.53	78.21

Table 1. Example 1: RMSE and CPU time averaged over 200 observation realizations.

	VRPF		**TDSMC**	
N	**RMSE / km**	**CPU / s**	**RMSE / km**	**CPU / s**
50	42.62	0.24	0.88	1.32
100	33.49	0.49	0.66	2.62
250	22.89	1.23	0.54	6.56
500	17.26	2.42	0.51	12.98
1000	12.68	5.00	0.50	26.07
2500	6.18	13.20	0.49	67.32
5000	3.52	28.79	0.48	142.84

Table 2. Example 2: RMSE and CPU time averaged over 200 observation realizations.

3.2. Example 2

The same motion model was used as in Example 1, but with gamma-distributed arrival times, with shape and scale parameters $a = 10$ and $b = 5\Delta/a$ respectively, corresponding to a mean inter-arrival time of 5Δ. Along the position trajectory 37 independent Gaussian range and bearing measurements were generated at intervals $\Delta = 5$s with standard deviations $\sigma_r = 500$m and $\sigma_b = 0.01$ rads, respectively. The sensor was located at the origin. The proposal kernel was

chosen as a mixture of adjustment and birth moves, in proportion $2:1$. Experiments showed that for filtering, the use of other moves did not lead to significant improvement in performance. Approximations to the optimal forward and backward kernel were obtained by local linearisation of the observation model. This approach is commonly used to approximate the optimal proposal distribution in standard particle filtering, see [10]. Resampling was applied in the same manner as in Example 1.

The results in table 2 show that for the same CPU time, and therefore fewer particles, the proposed algorithm significantly outperforms the VRPF for this more challenging model. In the case of this model the TDSMC algorithm has higher CPU cost per particle due to the fact that short inter-arrival times have low prior probability and computation of kernel parameters and weight expressions in the TDSMC algorithm is more complicated.

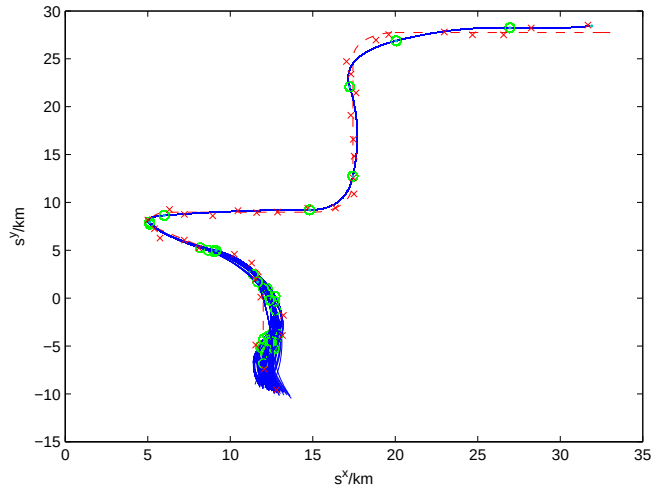

Fig. 2. Benchmark 2D position trajectory (dashed), observations (crosses), TDSMC particle position trajectories (solid) and jump locations (circles) after resampling at the final iteration.

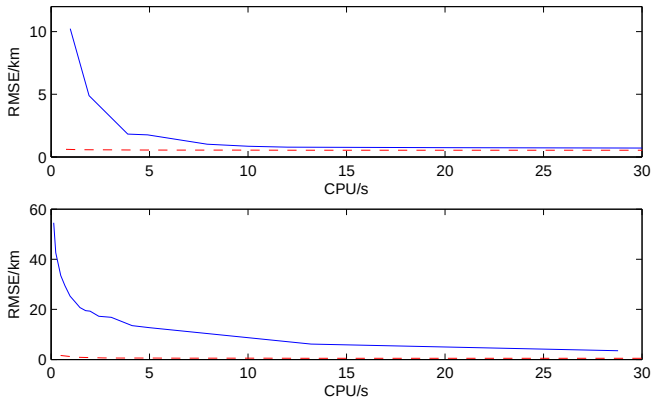

Fig. 3. RMSE vs average CPU time. Top: Example 1. Bottom: Example 2. VRPF (solid) and TDSMC (dashed).

Whilst estimates should not be made from the degenerate history of the particles, figure 2 gives an impression of the typical overall performance of the TDSMC algorithm using 500 particles and its ability to fit jump locations to observations. The advantage of the TDSMC algorithm in terms of RMSE against computational cost is summarized in figure 3.

4. CONCLUSION

We have presented a formulation of the filtering problem for a continuous time stochastic process observed at discrete points in time and have developed an inference scheme based on the framework of SMC samplers. The proposed approach treats the dimensionality of the problem explicitly and involves efficient particle proposal mechanisms. It out-performs existing methods. Future work will investigate schemes for adapting the mixture weights in the proposal kernel, so that they are matched to the observations.

5. REFERENCES

[1] P. Del Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society B*, vol. 63, no. 3, pp. 411–436, 2006.

[2] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science. Springer Verlag, New York, 2001.

[3] S.J. Godsill, J. Vermaak, K-F. Ng, and J-F. Li, "Models and algorithms for tracking of manoeuvring objects using variable rate particle filters," *Proc. IEEE*, April 2007, (To Appear).

[4] W.D. Blair, G.A. Watson, T. Kirubarajan, and Y. Bar-Shalom, "Benchmark for radar allocation and tracking in ECM," *IEEE Trans. AES*, vol. 34, no. 4, pp. 1097–1114, October 1998.

[5] S. J. Godsill and J. Vermaak, "Models and algorithms for tracking using trans-dimensional sequential Monte Carlo," in *Proc. IEEE ICASSP*, 2004.

[6] S. Maskell, "Joint tracking of manoeuvring targets and classification of their manoeuvrability," *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2339–2350, 2004.

[7] P. Del Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo methods for Bayesian Computation," in *Bayesian Statistics 8*. Oxford University Press, 2006.

[8] A. Doucet, L. Montesano, and A. Jasra, "Optimal filtering for partially observed point processes using trans-dimensional sequential Monte Carlo," in *Proc. IEEE ICASSP*, 2006.

[9] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, March 1994.

[10] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.