

Auxiliary Particle Implementation of the Probability Hypothesis Density Filter

Nick Whiteley, Sumeetpal Singh and Simon Godsill

Abstract

Optimal Bayesian multi-target filtering is, in general, computationally impractical owing to the high dimensionality of the multi-target state. The Probability Hypothesis Density (PHD) filter propagates the first moment of the multi-target posterior distribution. While this reduces the dimensionality of the problem, the PHD filter still involves intractable integrals in many cases of interest. Several authors have proposed Sequential Monte Carlo (SMC) implementations of the PHD filter. However, these implementations are the equivalent of the Bootstrap Particle Filter, and the latter is well known to be inefficient. Drawing on ideas from the Auxiliary Particle Filter (APF), we present a SMC implementation of the PHD filter which employs auxiliary variables to enhance its efficiency. Numerical examples are presented for two scenarios, including a challenging nonlinear observation model.

Index Terms

PHD Filter, Sequential Monte Carlo, Multi-Target Tracking, Auxiliary Particle Filter.

I. INTRODUCTION

Multi-target filtering is a dynamic state estimation problem in which both the number of hidden targets and the locations of the targets are unknown. Additionally, the targets appear and terminate at random times. The modelling of multi-target dynamics in this manner naturally incorporates track initiation and termination, a procedure that has mostly been performed separately in traditional tracking algorithms.

As in the single-target case, optimal multi-target filtering involves the propagation of the posterior distribution through Bayes' law. Exact optimal multi-target filtering is impossible in many cases of interest due to the presence of intractable integrals in the filtering recursion. The application of numerical methods (Monte Carlo or otherwise) to approximate the optimal filter for multi-target models is extremely computationally intensive owing to the high dimensionality of the multi-target state.

Consider the state space of a single target, $E \subset \mathbb{R}^d$. Each point in this space may specify, for example, the position and velocity of the target. Multi-target filtering involves recursive computation of a distribution over the

Nick Whiteley is with the Statistics Group, Department of Mathematics, University of Bristol, University Walk Bristol, BS8 1TW, UK.

Sumeetpal Singh and Simon Godsill are with the Signal Processing Laboratory, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.

number of targets and each of their locations in E , given a sequence of noisy observations. The multi-target posterior filtering distribution is therefore a probability distribution on the disjoint union $\biguplus_{k \geq 0} E^k$.

A more tractable alternative to the optimal multi-target filter is the Probability Hypothesis Density filter of Mahler, [1], which propagates the first moment of the multi-target posterior, known as the intensity function or PHD. A variety of related material and further information about the PHD filter can be found in [2][Chapter 16].

While multi-target tracking is a mature field, the direct application of Point Process (PP) formalism to the derivation of the PHD filter is new to the area, with recent developments being made in [3]. The PHD filter has recently been the focus of much interest due to its favourable performance in multi-target tracking compared to traditional approaches (we refer the reader to [4] for a variety of approaches to multi-target tracking not based on the PP formalism). A generalization of the scheme, which propagates a probability distribution on the number of hidden targets, known as the Cardinalized PHD (CPHD) filter, has also been proposed [5]. The motivation for the development of this technique was to improve the quality of estimates of the number of hidden targets. The multi-target tracking model to which the PHD filter can be applied has attracted significant attention in the aerospace literature. However, this underlying model has recently also found applications in other fields such as communications engineering, computer vision and audio signal processing, [6], [7], [8], [9], and therefore the PHD filter is applicable in all these scenarios.

Sequential Monte Carlo (SMC) methods, [10], are a family of powerful algorithms which recursively propagate a set of weighted random samples, termed *particles*, in order to approximate probability distributions of interest. SMC methods are motivated by the asymptotic properties of the particle set. Under weak assumptions, as the number of samples increases, the integral of a test function with respect to the distribution defined by the particle set converges to the integral of that function with respect to the corresponding true distribution. In the context of filtering such methods are known as *particle filters*. For non-linear, non-Gaussian state-space models the exact optimal filter is analytically intractable, but SMC methods can yield very good approximations of it. For such models, it has been widely documented that SMC methods can exhibit significantly better performance than deterministic sub-optimal filters such as the Extended Kalman filter (EKF), [11] and the Unscented Kalman filter (UKF) [12]. SMC methods can and have been applied to the full Bayesian multi-target tracking problem, for example see [13], [14], [15], [16], [17], but if targeting the true full Bayesian multi-target posterior distribution, such approaches become computationally very expensive in high-dimensions, which is the case when tracking several targets. Performance of such algorithms may be improved for a specific class of models in which it is possible to analytically integrate out part of the model (a procedure known as Rao-Blackwellisation), [18], but such an approach will ultimately still be affected by high dimensionality of the problem when the number of targets is large.

SMC methods can also be used to approximate the intensity function of a PP and have previously been employed to approximate the PHD recursion, [19], [20], [21], [18]. Instead of multi-target probability distributions, the set of weighted samples is now used to approximate intensity functions, including their total mass.

In this paper we present a new SMC implementation of the PHD filter which significantly out-performs algorithms proposed in the literature thus far. It builds on ideas from a specific SMC method originating from the Auxiliary

Particle Filter (APF) of Pitt and Shephard [22]. This method boosts the efficiency of the algorithm by pre-selecting particles for propagation on the basis of their relationship with the most recently received observations. The proposed method exhibits a natural mechanism for clustering of particles on the basis of the observations to which they are assigned. This can be used as a tool for extracting state estimates. A preliminary version of the algorithm appeared in [23].

The structure of this paper is as follows. In section II we describe the model underlying the PHD recursion, state the recursion itself and fix notation. In section III we describe SMC methods and the APF. Section IV describes existing particle implementations of the PHD filter. The proposed algorithm is formulated in section V where we provide expressions for optimal proposal distributions and interpret them in terms of the PP theory underlying the PHD recursion. Numerical results are presented in section VI, for two different models. We demonstrate methods for constructing proposal distributions and show the improvement in performance which is possible with the proposed methods.

II. THE PHD FILTER

The PHD filter was originally developed in the framework of Finite Set Statistics (FISST) [1]. The relationship between FISST, conventional probability and conventional point process theory is discussed in [24] and [2][Appendices E and F]. More recently, the multi-target tracking problem has been formulated and the PHD filter derived directly using the theory of PP's [3]. For the purposes of the present work we concern ourselves with the following constructive definition of a finite¹ PP [25]. A finite PP, $\mathbf{X} = X_{1:K}$, is a random number K of random points $X_{1:K}$ each valued in a state space E , for example, $E \subset \mathbb{R}^d$, so that the PP itself takes values in the disjoint union $\biguplus_{k \geq 0} E^k$. The probability distribution of \mathbf{X} can be specified by a probability distribution on the total number of points and, for each integer $k \geq 1$, a probability distribution on E^k , which determines the conditional joint probability distribution of the points $X_{1:k}$, given that their total number is k .

The first moment or intensity of a PP yields the expected number of points in a region of the state space. We will specify this first moment in terms of an *intensity function*, $\alpha : E \rightarrow \mathbb{R}_+$, so that:

$$\mathbb{E}[N(A)] = \int_A \alpha(x) dx, \quad A \in \mathcal{B}(E),$$

where $N(A)$ is the number of points of \mathbf{X} which are in the set A and $\mathcal{B}(E)$ is the Borel σ -algebra on E (for theoretical details, see [25]). In the context of multi-target tracking, each point of \mathbf{X} represents a hidden target. The intensity function of the posterior distribution of \mathbf{X} is therefore very useful because it yields the expected number of targets in any region of the state space. Peaks in the intensity function can be used to estimate target locations and the total mass of the intensity function provides an estimate of the total number of targets. A filtering scheme which propagates only this intensity function, as opposed to the full posterior, is attractive as the dimensionality of the problem is effectively reduced to the dimensionality of E .

¹For convenience we will drop the 'finite' prefix in what follows.

The PHD filter consists of a prediction and update operation which propagates the intensity function of the multi-target posterior recursively in time [1]. It is derived from the following model for the dynamics of the hidden targets and noisy observations. See [25], [26] for background theory on Point Processes.

Consider a spatial Poisson process of unobserved points $\mathbf{X}_1 = X_{1,1:K}$, where each element of \mathbf{X}_1 is random point in a state space E . Denote the intensity function of \mathbf{X}_1 by $\hat{\alpha}_1(x)$. With probability $p_S(x)$, each point of \mathbf{X} survives and mutates according to a Markov transition kernel on E with density $f(x_2|x_1)$. This happens independently for each point in \mathbf{X}_1 . In addition to the surviving points of \mathbf{X}_1 , new points are ‘born’ from a Poisson process with intensity function $\gamma(x)$. Denote by \mathbf{X}_2 the PP on E defined by the superposition of the surviving points of \mathbf{X}_1 and the newly born points. We denote it’s intensity function by $\alpha_2(x)$.

The points of \mathbf{X}_2 are observed through the following model. With probability $p_D(x)$, each point of \mathbf{X}_2 generates a noisy observation in an observation space F through a kernel with density $g(y|x)$. This happens independently for each point of \mathbf{X}_2 . Let Θ denote observations originating from \mathbf{X}_2 . In addition to these detected points, clutter points from an independent Poisson process on F , denoted by \mathbf{K} , with intensity function $\kappa(y)$, are also observed. Denote by \mathbf{Y} the superposition of Θ and \mathbf{K} , and a realization of \mathbf{Y} by $\mathbf{y} = y_{1:m}$.

By the application of Bayes’ rule, the posterior distribution of \mathbf{X}_2 given \mathbf{y} can be obtained. Characterization of this posterior distribution was first performed implicitly in [1] via probability generating functionals and later explicitly in [3]. This posterior distribution is not Poisson, but it can be shown that the Poisson process which is distributed most closely to it in the sense of Kullback-Leibler divergence must have the same first moment. It is this intensity function which is propagated by the PHD recursion, which has the following prediction/update structure at its n th iteration:

$$\alpha_n(x_n) = \int_E f(x_n|x_{n-1})p_S(x_{n-1})\hat{\alpha}_{n-1}(x_{n-1})dx_{n-1} + \gamma(x_n), \quad (1)$$

$$\hat{\alpha}_n(x_n) = \left[1 - p_D(x_n) + \sum_{p=1}^{m_n} \frac{\psi_{n,p}(x_n)}{\mathcal{Z}_{n,p}} \right] \alpha_n(x_n), \quad (2)$$

where for $p = 1, 2, \dots, m_n$, suppressing the dependence on the observed quantities for convenience,

$$\begin{aligned} \psi_{n,p}(x) &= p_D(x)g(y_{n,p}|x), \\ \mathcal{Z}_{n,p} &= \int_E \psi_{n,p}(x)\alpha_n(x)dx + \kappa(y_{n,p}). \end{aligned}$$

In this notation, $\alpha_n(x)$ and $\hat{\alpha}_n(x)$ are respectively termed the predicted and updated intensities at iteration n . We denote by $g(y_{n,p}|x)$ the likelihood for the p th observation at iteration n and by m_n the total number of observations at iteration n . In the following we also adopt the notation $\mathbb{M}_n = \{1, 2, \dots, m_n\}$.

To aid presentation later on, we will employ the following notation in describing the PHD recursion on an extended state space $E' = E \cup \{s\}$, where s is an isolated point that does not belong to E , and is termed a ‘source’ state. Abusing notation we denote by x either a point in E or E' depending on the context. The extended recursion is as follows:

$$\alpha_n(x_n) = \int_{E'} f'(x_n|x_{n-1})p'_S(x_{n-1})\widehat{\alpha}'_{n-1}(x_{n-1})dx_{n-1}, \quad (3)$$

$$\widehat{\alpha}_n(x_n) = \left[1 - p_D(x) + \sum_{p=1}^{m_n} \frac{\psi_{n,p}(x_n)}{\mathcal{Z}_{n,p}} \right] \alpha_n(x_n), \quad (4)$$

$$\widehat{\alpha}'_n(x_n) = \widehat{\alpha}_n(x_n) + \Gamma\delta_s(x_n), \quad (5)$$

where $\widehat{\alpha}'_n(x_n)$ is the extended updated intensity on E' and $\Gamma = \int_E \gamma(x)dx$. The transition density, $f'(x_n|x_{n-1})$ is extended to act from E' to E as follows:

$$f'(x_n|x_{n-1}) = \begin{cases} f(x_n|x_{n-1}) & x_{n-1} \in E \\ \gamma(x_n)/\Gamma & x_{n-1} = s \end{cases},$$

and the survival probability extended to E' as:

$$p'_S(x) = \begin{cases} p_S(x) & x \in E \\ 1 & x = s \end{cases}.$$

The specification of the PHD recursion as per (3)–(5) is of interest because (as we shall see in Section V) it allows one iteration of the PHD predict/update operation to be written in a fashion which can be approximated directly using auxiliary SMC techniques. The idea of the ‘source’ state will allow the predicted intensity from the previous iteration and birth intensity to be dealt with in a unified sampling scheme, due to the form of (3).

Whilst the PHD filter reduces the dimensionality of the problem, the PHD recursion still involves intractable integrals in many cases of interest, the exception being the ‘linear–Gaussian’ case, where the PHD has a Gaussian mixture form, [27]. We subsequently refer to this case as the GM-PHD filter. However, in this case the number of mixture components increases over time. Therefore, in order to bound the computational cost per iteration of the filtering algorithm, a pruning/merging technique is typically employed [27]. The error arising from this pruning/merging was quantified in [28]. One practical advantage of the GM-PHD filter is that it readily admits an heuristic for state–estimation: estimates can be extracted directly from the means of the Gaussian mixture components, which (post–pruning) correspond to local, but significant maxima of the intensity function.

PHD filter analogs of the EKF and UKF have also been developed for use in non–linear models. These methods rely on some deterministic approximation, (local linearization of the model in case of the EKF and weighted point approximations of moments in the case of the UKF) and as such will always exhibit some degree of bias relative to the corresponding true PHD recursion. However, in mildly nonlinear scenarios they have been shown to perform well in practice and can be computationally inexpensive [27].

As introduced in Section I, the CPHD filter propagates a probability distribution over the number of hidden targets in addition to the PHD itself. It has been demonstrated that this approach can produce more reliable estimates of the number of hidden targets. Furthermore, it can be combined with the Gaussian mixture and EKF/UKF techniques described above [29].

In general non-linear scenarios, numerical methods which permit an arbitrarily close approximation of the true PHD are attractive. Monte Carlo methods are one class of such methods and the next section introduces some of their characteristics.

III. SEQUENTIAL MONTE CARLO

Sequential Monte Carlo methods have become a standard tool for computation in non-linear optimal filtering problems and in this context have been termed *particle filters*. We do not give explicit details of standard particle filtering algorithms here, but rather provide the following summary and refer to [10] and [30] for a variety of algorithms, theoretical details and applications, [31] for a general framework and [32] for underlying Monte Carlo methods. SMC algorithms may be viewed as being constructed from ideas of Sequential Importance Sampling (SIS) and resampling. They recursively propagate a set of weighted random samples called *particles*, which are used to approximate a sequence of probability distributions. The algorithms are such that, as the number of particles tends to infinity and under weak assumptions, an integral with respect to the distribution defined by the particle set converges to the integral with respect to the corresponding true distribution.

A typical SMC algorithm consists of recursively proposing samples from some instrumental distribution and computing importance weights which account for the discrepancy between the instrumental distribution and the distribution of interest. Occasionally, resampling from the distribution defined by the particle set is performed. This resampling step involves duplicating particles with high weights and discarding particles with low weights. It is crucial in stabilizing the algorithm, as without it an SIS scheme would rapidly degenerate, carrying all weight on a single particle and thus becoming useless for online estimation. Estimates drawn from such an algorithm would typically have very high variance. Treatments of this issue from a theoretical point of view are given in [33], [34] and a comprehensive study of the underlying theory can be found in the book length volume [35].

As mentioned in the introduction, SMC methods can be computationally very expensive when targeting the full Bayesian multi-target posterior because of the high-dimension of the space on which the filtering distribution is defined. This expense arises from the need to use more particles to combat an increase in estimator variance which is, in essence, caused by the same phenomenon of weight degeneracy which occurs when applying sequential importance sampling in high dimensions (albeit occurring at single iterations of the algorithm).

A generic and practical strategy by which to achieve efficiency in an SMC scheme is to ensure that the variance of the importance weights is minimised [36]. This is important even in low dimensions. If the variance of the importance weights is large, despite resampling, the algorithm will suffer from the same problem of importance weight degeneracy. Therefore an important factor in the practical efficiency of SMC methods is the mechanism by which particles are proposed. If degeneracy of the weights is to be avoided, this mechanism should take into account information from the observations and drive particles into regions of high probability under the target distribution. The *Bootstrap* particle filter, [37], proposes particles from the transition kernel of the hidden process in the state-space model, and resamples at every iteration. As it does not take into account information from the observations, it is typically inefficient and large numbers of particles are required if reliable estimates are to be

obtained from it.

The Auxiliary Particle Filter (APF) of Pitt and Shephard, [22], selects particles for propagation on the basis of how well matched they are to the observation at the next time step, with the aim of reducing the variance of the importance weights. The method originally took its name from a set of auxiliary variables introduced to index the particle set and in [38] it was noted that the scheme has a proposal distribution of the same form as standard particle filtering algorithms. More recently, in Johansen and Doucet, [39], the APF was reinterpreted as a sequential importance sampling/resampling (SISR) algorithm targeting an auxiliary sequence of distributions, with importance weights correcting for the discrepancy between these distributions and the distributions of interest. Thus the selection of particles arises simply from applying resampling to a particle set subject to a set of auxiliary importance weights. In the following sections we present an algorithm which applies ideas from the APF to the implementation of the PHD filter.

IV. PARTICLE PHD FILTER

A particle implementation of the PHD filter in its full generality was proposed in [19], around the same time as two other independent works [20] and [21]. In [20], only the special case without clutter was considered. On the other hand, [21] describes an implementation for the special case with neither birth nor spawning. The theme common to these approaches is the propagation of a particle approximation to the intensity function through the PHD recursion (1) and (2).

One iteration of existing particle PHD filters is outlined as follows. Samples are drawn from a proposal distribution, conditionally upon the previous particle set, and weighted in accordance with the prediction operation. Supplementary particles dedicated to the birth term are then added. This yields an approximation to the predicted intensity α_n , which is used in its entirety to approximate the integrals of the form $\int_E \psi_{n,p}(x)\alpha_n(x)dx$ in the denominators of (2). The particles are then re-weighted according to the update operator and resampling is performed. The total weight of the particle set must be maintained and it yields an estimate of total number of targets in E .

In this framework it is not obvious how to choose the proposal distribution in order to minimise the variance of the weights. In practice, proposals are often made from the prior, which is sub-optimal, and this is the analogue of the Bootstrap particle filter, which is inefficient.

The particle implementation scheme of [21] includes regularization. This idea involves resampling not from the weighted particle set, but from an absolutely continuous probability distribution defined by convolving the particle set (or subsets thereof, obtained by clustering) with a kernel. This idea has been applied to SMC algorithms in the context of optimal filtering with the aim of increasing diversity of particle locations and the stability over time of the algorithm, see for example [37], [40] and references therein. It should be noted that regularization does not employ information from the next observation to guide the selection or proposal of particles, in contrast to the method proposed below. We note that it is possible to combine regularization and local resampling with the proposed scheme, although we do not explore this idea further here.

Convergence results establishing the theoretical validity of the particle PHD filter have been obtained in [24], [41] and [42].

V. AUXILIARY PARTICLE PHD FILTER

A. Outline of the Approach

In the APF, particles are selected for propagation on the basis of how well they explain the next observation. We employ this idea but with an extra selection step that first selects the observations that are explained by the particle set as a whole. We will perform sampling on a higher dimensional space than existing implementations, but doing so will allow us to:

- (A1) Manage efficiently a fixed particle budget by concentrating effort on those observations which most probably originate from true targets and therefore contribute most significantly to the r.h.s. of (2).
- (A2) Employ a particle selection scheme which takes into account observations, as in the APF.
- (A3) Tailor the distributions from which samples are drawn to the observations using existing techniques from the SMC literature.

To pave the way towards the proposed algorithm, we next consider an hypothetical, static IS problem which is modified in the next section to yield a recursive algorithm.

Consider the problem of estimating the following integral:

$$\bar{\varphi} = \int_E \varphi(x_n) \hat{\alpha}_n(x_n) dx_n, \quad (6)$$

where φ is some suitable test function on E . Assuming (for now) that the normalizing constants $\{\mathcal{Z}_{n,p}\}_{p=1}^{m_n}$, are known, a Monte Carlo estimate of (6) may hypothetically be built using the following importance sampling (IS) identity on $E \times E' \times \mathbb{M}_n$ and $E \times E'$. This is just a re-writing of (1) and (2), using the notation on the extended state space E' , (3)–(5):

$$\begin{aligned} \bar{\varphi} = & \sum_{p=1}^{m_n} \int_E \int_{E'} \varphi(x_n) \frac{\psi_{n,p}(x_n)}{\mathcal{Z}_{n,p}} \frac{f'(x_n|x_{n-1}) p'_S(x_{n-1}) \hat{\alpha}'_{n-1}(x_{n-1})}{q_n^{(1)}(x_n, x_{n-1}, p)} q_n^{(1)}(x_n, x_{n-1}, p) dx_{n-1} dx_n \\ & + \int_E \int_{E'} \varphi(x_n) \frac{[1 - p_D(x_n)] f'(x_n|x_{n-1}) p'_S(x_{n-1}) \hat{\alpha}'_{n-1}(x_{n-1})}{q_n^{(2)}(x_n, x_{n-1})} q_n^{(2)}(x_n, x_{n-1}) dx_{n-1} dx_n, \end{aligned} \quad (7)$$

where $q_n^{(1)}(x_n, x_{n-1}, p)$ and $q_n^{(2)}(x_n, x_{n-1})$ are probability densities on $E \times E' \times \mathbb{M}_n$ and $E \times E'$ respectively.

The idea behind the proposed SMC algorithm is to target (7) using IS. We note that there are several ways in which the integral (6) could be decomposed and targeted using IS. For example, one could attempt to evaluate the summation in the first term on the r.h.s. of (7) analytically, as opposed to sampling over observation indices. However, the decomposition in (8) has been chosen for a specific reason related to point (A1). The discussion to follow expands on this.

We focus on proposal distributions of the following form, assumed to have support such that (7) is valid:

$$q_n^{(1)}(x_n, x_{n-1}, p) = q_n^{(1)}(x_n | x_{n-1}, p) \frac{V_{n,p}^{(1)}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1})}{\int_{E'} V_{n,p}^{(1)}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}} q_n^{(1)}(p),$$

$$q_n^{(2)}(x_n, x_{n-1}) = q_n^{(2)}(x_n | x_{n-1}) \frac{V_n^{(2)}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1})}{\int_{E'} V_n^{(2)}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}},$$

where for each p , $V_{n,p}^{(1)} : E' \rightarrow \mathbb{R}^+$ and $V_n^{(2)} : E' \rightarrow \mathbb{R}^+$ are bounded potential functions. Each of these functions is related to the concept of the ‘first-stage’ weight in the APF, whose calculation involves evaluating a potential function at sample locations. This parallel is made more apparent in the sequel. The rationale behind this approach is that it is possible to achieve performance which is superior to that of existing SMC implementations of the PHD filter by building a proposal distribution in terms of the updated intensity (or its particle approximation) from the previous time step.

We can then draw $N^{(1)}$ independent, identically distributed (iid) samples, $\{X_{n,1}^{(i)}, X_{n-1,1}^{(i)}, P_{n,1}^{(i)}\}_{i=1}^{N^{(1)}}$ from $q_n^{(1)}(x_n, x_{n-1}, p)$, $N^{(2)}$ iid samples, $\{X_{n,2}^{(i)}, X_{n-1,2}^{(i)}\}_{i=1}^{N^{(2)}}$ from $q_n^{(2)}(x_n, x_{n-1})$ and form the following estimator of $\bar{\varphi}$:

$$\bar{\varphi} = \sum_{i=1}^{N^{(1)}} \varphi(X_{n,1}^{(i)}) w_n^{(1)}(X_{n,1}^{(i)}, X_{n-1,1}^{(i)}, P_{n,1}^{(i)}) + \sum_{i=1}^{N^{(2)}} \varphi(X_{n,2}^{(i)}) w_n^{(2)}(X_{n,2}^{(i)}, X_{n-1,2}^{(i)}) \quad (8)$$

where the importance weights are given by:

$$w_n^{(1)}(x_n, x_{n-1}, p) = \frac{1}{N^{(1)}} \frac{1}{\mathcal{Z}_{n,p}} \frac{\psi_{n,p}(x_n) f'(x_n | x_{n-1}) p'_S(x_{n-1}) \alpha'_{n-1}(x_{n-1})}{q_n^{(1)}(x_n, x_{n-1}, p)} \quad (9)$$

$$= \frac{1}{N^{(1)}} \frac{1}{\mathcal{Z}_{n,p}} \frac{\int_{E'} V_{n,p}^{(1)}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}}{q_n^{(1)}(p)} \\ \times \frac{\psi_{n,p}(x_n) f'(x_n | x_{n-1}) p'_S(x_{n-1})}{q_n^{(1)}(x_n | x_{n-1}, p) V_{n,p}^{(1)}(x_{n-1})}, \quad (10)$$

and

$$w_n^{(2)}(x_n, x_{n-1}) = \frac{1}{N^{(2)}} \frac{[1 - p_D(x_n)] f'(x_n | x_{n-1}) p'_S(x_{n-1}) \alpha'_{n-1}(x_{n-1})}{q_n^{(2)}(x_n, x_{n-1})} \quad (11)$$

$$= \frac{1}{N^{(2)}} \int_{E'} V_n^{(2)}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1} \\ \times \frac{[1 - p_D(x_n)] f'(x_n | x_{n-1}) p'_S(x_{n-1})}{q_n^{(2)}(x_n | x_{n-1}) V_n^{(2)}(x_{n-1})}. \quad (12)$$

1) *Optimal Proposal Distributions:* As previously described, a generic scheme for ensuring that our estimator does not have high variance is to minimise the variance of the importance weights. The following proposition establishes the choice of proposal distributions which are optimal in this sense.

Proposition 1: The following choices minimise the variance of the importance weights, $w_n^{(1)}(X_n, X_{n-1}, P_n)$ and $w_n^{(2)}(X_n, X_{n-1})$:

$$\begin{aligned}
q_n^{(1),\text{opt}}(p) &\propto \frac{1}{\mathcal{Z}_{n,p}} \int V_{n,p}^{(1),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}, \\
V_{n,p}^{(1),\text{opt}}(x_{n-1}) &= \int_E \psi_{n,p}(x_n) f'(x_n|x_{n-1}) p'_S(x_{n-1}) dx_n, \\
q_n^{(1),\text{opt}}(x_n|x_{n-1}, p) &= \frac{\psi_{n,p}(x_n) f'(x_n|x_{n-1}) p'_S(x_{n-1})}{\int_E \psi_{n,p}(x_n) f'(x_n|x_{n-1}) p'_S(x_{n-1}) dx_n}, \\
V_n^{(2),\text{opt}}(x_{n-1}) &= \int_E [1 - p_D(x_n)] f'(x_n|x_{n-1}) p'_S(x_{n-1}) dx_n, \\
q_n^{(2),\text{opt}}(x_n|x_{n-1}) &= \frac{[1 - p_D(x_n)] f'(x_n|x_{n-1}) p'_S(x_{n-1})}{\int_E [1 - p_D(x_n)] f'(x_n|x_{n-1}) p'_S(x_{n-1}) dx_n}, \tag{13}
\end{aligned}$$

and in this case the estimator becomes:

$$\begin{aligned}
\tilde{\varphi}_n &= \frac{\sum_{p=1}^{m_n} \frac{1}{\mathcal{Z}_{n,p}} \int_{E'} V_{n,p}^{(1),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}}{N^{(1)}} \sum_{i=1}^{N^{(1)}} \varphi(X_{n,1}^{(i)}) \\
&\quad + \frac{\int_{E'} V_n^{(2),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \varphi(X_{n,2}^{(i)}).
\end{aligned}$$

Proof: Substitution readily yields that in the case of employing the optimal proposal distributions, the variances of the weights (9) and (11) are both zero, as the importance weights are of the form:

$$w_n^{(1)}(x_n, x_{n-1}, m) = \frac{1}{N^{(1)}} \sum_{p=1}^{m_n} \frac{1}{\mathcal{Z}_{n,p}} \int_{E'} V_{n,p}^{(1),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1},$$

and

$$w_n^{(2)}(x_n, x_{n-1}) = \frac{1}{N^{(2)}} \int_{E'} V_n^{(2),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1},$$

which do not depend on x_n , x_{n-1} or m . ■

Corollary 1: For a total of $N = N^{(1)} + N^{(2)}$ samples, and when the optimal proposal distributions are employed, the importance weights are of zero variance and are all equal in the case that:

$$\begin{aligned}
N^{(1)} &\propto N \times \sum_{p=1}^{m_n} \frac{1}{\mathcal{Z}_{n,p}} \int_{E'} V_{n,p}^{(1),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1} \\
N^{(2)} &\propto N \times \int_{E'} V_n^{(2),\text{opt}}(x_{n-1}) \widehat{\alpha}'_{n-1}(x_{n-1}) dx_{n-1}
\end{aligned}$$

Note that the optimal choice of proposal distributions correspond to denominators of the two rightmost terms in (10) and (12) being matched to their respective numerators. This is precisely as in the APF and has the following interpretation. Given p , we choose the particle from time $n - 1$ that has a high weight and can explain $y_{n,p}$ well. This particle is then propagated forward via the optimal proposal for this observation, which is $q_n^{(1),\text{opt}}(x_n|x_{n-1}, p)$. Similarly if we match $q_n^{(1)}(p)$ to the first term on the r.h.s. of (10), the weight contribution of that term will be independent of p . Choosing $q_n^{(1)}(p)$ in this manner, as formalised below, amounts to selecting observations according

to the strength of the hypothesis that they are target generated. More specifically, the optimal proposal distribution, $q_n^{(1),\text{opt}}(p)$, has a probabilistic interpretation in terms of the spatial PP theory underlying the PHD recursion, see [3]. For convenience we reproduce the relevant aspects of the model from which the the PHD recursion is derived.

Consider a spatial Poisson process $\mathbf{X} = X_{1:K}$, where each X is random point in a state space E . Denote the intensity of \mathbf{X} by $\alpha_n(x)$ and consider the following observation model. With probability $p_D(x)$, each point of \mathbf{X} generates a noisy observation in an observation space F through a kernel with density $g(y|x)$. This happens independently for each point in \mathbf{X} . Let Θ denote the set of observations generated by points in \mathbf{X} . In addition to these detected points, clutter points from an independent Poisson process on F , denoted by \mathbf{K} , with intensity $\kappa(y)$, are also observed. Denote by \mathbf{Y} the superposition of Θ and \mathbf{K} .

Proposition 2: Let \mathbf{X} be a Poisson process as above, with intensity $\alpha_n(x)$. Then given a realization of observations, $\mathbf{y}_n = y_{n,1:m_n}$, according to the model described above, $q_n^{(1),\text{opt}}(p)$ as defined in (13) is proportional to the the posterior probability that the p th observed point, $y_{n,p}$, originates from a hidden target.

Proof: The proof is adapted from [3] and involves the concept of marking a Poisson process, see [26][pp. 55] for a definition. Let $\widehat{\mathbf{X}}$ be the points of \mathbf{X} that generate observations. Because points of \mathbf{X} generate observations independently with location dependent probability $p_D(x)$, it follows that $\widehat{\mathbf{X}}$ is a Poisson process with intensity $p_D(x)\alpha_n(x)$ [26][pp. 56]. Let \mathbf{Z} be a PP on $E \times F$ comprised of points (X, Y) , with the set of first coordinates given by the restriction of \mathbf{X} to points that generate observations, i.e., $\widehat{\mathbf{X}}$, and the second coordinates given by the corresponding observations. By the Marking Theorem [26][pp. 55], \mathbf{Z} is Poisson with intensity $p_D(x)\alpha_n(x)g(y|x)$.

We now define three new marked PP's with position space F and mark space $\check{E} = E \cup \{\Delta\}$ where $\{\Delta\}$ is an isolated one-point set which is not an element of E .

Firstly, define Ξ_1 to be a marked PP, where the point positions of Ξ_1 are given by \mathbf{K} (the clutter process), and every point in Ξ_1 carries Δ as its mark. It follows that Ξ_1 is Poisson on $F \times \{\Delta\}$. Secondly, define Ξ_2 to be $\{(Y, X) : (X, Y) \in \mathbf{Z}\}$. Since \mathbf{Z} is Poisson, the process defined by swapping its first and second coordinates is also a Poisson process. Thirdly, denote by Ξ the superposition of Ξ_1 and Ξ_2 . The superposition of two independent Poisson processes also yields a Poisson process with intensity being the sum of the two [26]. Thus Ξ is Poisson on $F \times \check{E}$ with intensity $v(y, x)$ given by:

$$v(y, x) = \begin{cases} g(y|x)p_D(x)\alpha_n(x) & x \in E, \\ \kappa(y) & x = \Delta, \end{cases}$$

with the abuse of notation that x is a point in \check{E} . Note that the second coordinate of a point of a Poisson process on a product space can be viewed as a mark. The mark distribution $\mu(x|y)$ for Ξ , which is the conditional distribution of the mark of a point, given its location y , is given by:

$$\mu(x|y) = \frac{v(y, x)}{\mathcal{Z}_n(y)},$$

where,

$$\mathcal{Z}_n(y) = \int g(y|x)p_D(x)\alpha_n(x)dx + \kappa(y).$$

The mark distribution is to be interpreted as follows: with probability $1 - \kappa(y)/\mathcal{Z}_n(y)$ a mark in E is generated according to the probability density $g(y|x)p_D(x)\alpha_n(x)/\int g(y|x)p_D(x)\alpha_n(x)dx$. Otherwise the mark Δ is chosen. The proof is then complete since conditioned on \mathbf{y}_n , the marks are independent and distributed according to $\prod_{p=1}^{m_n} \mu(x|y_{n,p})$, and therefore the posterior probability that an observed point $y_{n,p}$ originates from \mathbf{X} is given by:

$$\begin{aligned} 1 - \frac{\kappa(y_{n,p})}{\mathcal{Z}_n(y_{n,p})} &= \frac{\int_E g(y_{n,p}|x)p_D(x)\alpha_n(x)dx}{\int_E g(y_{n,p}|x)p_D(x)\alpha_n(x)dx + \kappa(y_{n,p})} \\ &\propto q_n^{(1),\text{opt}}(p). \end{aligned}$$

■

Thus sampling from $q_n^{(1),\text{opt}}(p)$ allocates more particles to those observations which have a high posterior probability of originating from hidden targets, achieving point (A1).

2) *Variance Reduction*: The variance of the estimator (8) can be reduced by the application of alternative methods to sample from $q_n^{(1)}(p)$. For example, the residual, stratified and systematic sampling mechanisms (commonly used to resample from the particle set in SMC methods) could be applied. See [43], [44], and references therein. These methods are adopted in section VI.

B. Proposed Algorithm

To progress from the IS identity (7) to a practical recursive algorithm targeting the sequence of updated intensity functions $\{\hat{\alpha}_n(x_n)\}_{n \geq 0}$, we need some modifications.

Assume that at time $n - 1$ we have available a particle approximation to $\hat{\alpha}_{n-1}$, denoted $\hat{\alpha}_{n-1}^N(dx_{n-1}) = \sum_{i=1}^N W_{n-1}^{(i)} \delta_{X_{n-1}^{(i)}}(dx_{n-1})$, in the sense that for some suitable test function φ :

$$\int_E \varphi(x) \hat{\alpha}_{n-1}^N(dx) \xrightarrow[N \rightarrow \infty]{a.s.} \int_E \varphi(x) \hat{\alpha}_{n-1}(dx). \quad (14)$$

Also denote the extended particle approximation on E' by:

$$\hat{\alpha}_{n-1}^{N'}(dx_{n-1}) = \hat{\alpha}_{n-1}^N(dx_{n-1}) + \Gamma \delta_s(dx_{n-1})$$

We modify the target integral (7) by replacing $\hat{\alpha}_{n-1}$ with its particle approximation. The proposal distributions are also modified, as follows:

$$\begin{aligned} q_n^{(1)}(dx_{n-1}|p) &= \frac{\sum_{i=1}^N V_{n,p}^{(1)}(x_{n-1}^{(i)}) w_{n-1}^{(i)} \delta_{x_{n-1}^{(i)}}(dx_{n-1}) + V_{n,p}^{(1)}(s) \Gamma \delta_s(dx_{n-1})}{\sum_{i=1}^N V_{n,p}^{(1)}(x_{n-1}^{(i)}) w_{n-1}^{(i)} + V_{n,p}^{(1)}(s) \Gamma}, \\ q_n^{(2)}(dx_{n-1}) &= \frac{\sum_{i=1}^N V_n^{(2)}(x_{n-1}^{(i)}) w_{n-1}^{(i)} \delta_{x_{n-1}^{(i)}}(dx_{n-1}) + V_n^{(2)}(s) \Gamma \delta_s(dx_{n-1})}{\sum_{i=1}^N V_n^{(2)}(x_{n-1}^{(i)}) w_{n-1}^{(i)} + V_n^{(2)}(s) \Gamma}. \end{aligned}$$

Thus sampling from $q_n^{(1)}(dx_{n-1}|p)$ and $q_n^{(2)}(dx_{n-1})$ now amounts to resampling from the union of the re-weighted particle set and an atom representing the birth mass. This is analogous to first stage sampling in the APF, achieving point (A2). Intuitively, once p is chosen, a particle from time $n-1$ is chosen to be propagated forward on the basis of its current weight and how well it explains $y_{n,p}$, taking into consideration survival and detection probabilities. Comments on the practical choices of other elements of the proposal distributions are made below.

In the hypothetical IS integral (7), it was assumed that the normalizing constants $\{\mathcal{Z}_{n,p}\}_{p=1}^{m_n}$ were known. In practice this is not the case, and we need to estimate them too. For each $\mathcal{Z}_{n,p}$, this can be achieved using the $N_p^{(1)}$ particles were sampled from $q_n^{(1)}(x_n, x_{n-1}|p)$. In principle, we could alternatively estimate $\mathcal{Z}_{n,p}$ using the other particles as well, but when the observations are well separated this will be inefficient as such particles would make very little contribution to the estimate. From the definition of $\mathcal{Z}_{n,p}$ we therefore construct the following IS identity:

$$\mathcal{Z}_{n,p} = \int_{E'} \int_E \frac{\psi_{n,p}(x_n) f'(x_n|x_{n-1}) p'_S(x_{n-1}) \widehat{\alpha}_{n-1}'(x_{n-1})}{q_n^{(1)}(x_n, x_{n-1}|p)} q_n^{(1)}(x_n, x_{n-1}|p) dx_n dx_{n-1} + \kappa(y_{n,p}),$$

and the corresponding estimator:

$$\tilde{\mathcal{Z}}_{n,p} = \frac{\int V_{n,p}^{(1)}(x) \widehat{\alpha}_{n-1}'(dx)}{N_p^{(1)}} \sum_{i \in \mathbb{T}_{n,p}} \frac{\psi_{n,p}(X_{n,1}^{(i)}) f'(X_{n,1}^{(i)}|X_{n-1,1}^{(i)}) p'_S(X_{n-1,1}^{(i)})}{q_n^{(1)}(X_{n,1}^{(i)}|X_{n-1,1}^{(i)}, p) V_{n,p}^{(1)}(X_{n-1,1}^{(i)})} + \kappa(y_{n,p}) \quad (15)$$

where $\mathbb{T}_{n,p} = \{i : P_{n,1}^{(i)} = p\}$, $N_p^{(1)} = \text{card}(\mathbb{T}_{n,p})$. This estimate of the normalizing constant is used in the place of its true value in the importance weights, (10), (12), which become:

$$w_n^{(1)}(x_n, x_{n-1}, p) = \frac{1}{N^{(1)}} \frac{1}{\tilde{\mathcal{Z}}_{n,p}} \frac{\int_{E'} V_{n,p}^{(1)}(x_{n-1}) \widehat{\alpha}_{n-1}'(dx_{n-1})}{q_n^{(1)}(p)} \times \frac{\psi_{n,p}(x_n) f'(x_n|x_{n-1}) p'_S(x_{n-1})}{q_n^{(1)}(x_n|x_{n-1}, p) V_{n,p}^{(1)}(x_{n-1})}, \quad (16)$$

and

$$w_n^{(2)}(x_n, x_{n-1}) = \frac{1}{N^{(2)}} \int_{E'} V_n^{(2)}(x_{n-1}) \widehat{\alpha}_{n-1}'(dx_{n-1}) \times \frac{[1 - p_D(x_n)] f'(x_n|x_{n-1}) p'_S(x_{n-1})}{q_n^{(2)}(x_n|x_{n-1}) V_n^{(2)}(x_{n-1})}. \quad (17)$$

Note that in practice we need not compute (15) for any p such that $N_p^{(1)} = 0$.

C. Design of Proposal Distributions

The optimal choices, $V_{n,p}^{(1),\text{opt}}(x_{n-1})$, $q_n^{(1),\text{opt}}(x_n|x_{n-1}, p)$ and $V_n^{(2),\text{opt}}(x_{n-1})$, $q_n^{(2),\text{opt}}(x_n|x_{n-1})$, may often not be available in practice.

The fact that the sampling scheme conditions on observation indices means that $q_n^{(1),\text{opt}}(x_n|x_{n-1}, p)$ is of the form:

$$q_n^{(1),\text{opt}}(x_n|x_{n-1}, p) \propto \psi_{n,p}(x_n) f'(x_n|x_{n-1}).$$

In order to approximate $q_n^{(1),\text{opt}}(x_n|x_{n-1},p)$ it is possible to use techniques commonly applied in conjunction with the standard particle filter, such as local-linearization of the dynamical model, Laplace approximations or the Unscented Transform (UT). Therefore point (A3) is achieved. See [36], [45] for several possibilities and section VI for specific details of use of the UT.

One possible strategy for choosing $V_{n,p}^{(1)}(x_{n-1})$, originating from [22], is to approximate $\int \psi_{n,p}(x_n)f'(x_n|x_{n-1})dx_n$ with $\psi_{n,p}(\mu_n(x_{n-1}))$, where $\mu_n(x_{n-1})$ is the mean or mode of $f'(x_n|x_{n-1})$. However, it was pointed out in [39] that such an approach can lead to an estimator with high variance, because the approximation may be too concentrated relative to $V_{n,p}^{(1),\text{opt}}(x_{n-1})$. To address this problem we could apply tempering to the particle set once it has been re-weighted by the potential function and set:

$$q_n^{(1)}(dx_{n-1}|p) = \frac{\sum_{i=1}^N [V_{n,p}^{(1)}(x_{n-1}^{(i)})w_{n-1}^{(i)}]^\epsilon \delta_{x_{n-1}^{(i)}}(dx_{n-1}) + [V_{n,p}^{(1)}(s)\Gamma]^\epsilon \delta_s(dx_{n-1})}{\sum_{i=1}^N [V_{n,p}^{(1)}(x_{n-1}^{(i)})w_{n-1}^{(i)}]^\epsilon + [V_{n,p}^{(1)}(s)\Gamma]^\epsilon}$$

where $0 < \epsilon < 1$. Similar techniques can be applied to approximate $V_n^{(2),\text{opt}}(x_{n-1})$. Other suggestions can be found in [39], [45].

The optimal choice of distribution over observation indices, $q_n^{(1)}(p)$, may not be available in practice (outside the ‘linear-Gaussian case’), as it involves the unknown normalizing constants, $\{\mathcal{Z}_{n,p}\}_{p=1}^{m_n}$, which are to be estimated *after* having sampled from $q_n^{(1)}(p)$. However, sensible approximation of this optimal distribution can still yield good performance. For example, having used one of the above methods to approximate the $V_{n,p}^{(1),\text{opt}}(x_{n-1})$, we could choose:

$$q_n^{(1)}(p) \propto \frac{\int_{E'} V_{n,p}^{(1)}(x) \hat{\alpha}_{n-1}^{N'}(dx)}{\int_{E'} V_{n,p}^{(1)}(x) \hat{\alpha}_{n-1}^{N'}(dx) + \kappa(y_{n,p})}. \quad (18)$$

The same issue of high estimator variance due to over-concentrated proposal distributions may arise here, and in practice a similar tempering approach may be necessary.

The algorithm for the auxiliary particle PHD filter is given below, with k_0 being the expected initial number of targets. The computational complexity of the algorithm is $\mathcal{O}(m_n N)$ at the n th iteration.

VI. SIMULATION STUDY

We present simulation results to demonstrate the improvements in performance over existing PHD filtering algorithms which are possible under the proposed scheme.

Consider a constant velocity tracking model for a vehicle whose position is specified in two dimensions, restricted to the window $[0, 100] \times [0, 100]$. The state of a single target is specified by a 4 dimensional vector $x_n = [x_{n,1} \ x_{n,2}; x_{n,3} \ x_{n,4}]^T$; $[x_{n,1} \ x_{n,3}]^T$ specifies position and $[x_{n,2} \ x_{n,4}]^T$ specifies velocity. The target dynamics are defined by:

```

1:  $n = 0$ 
2: for  $i = 1$  to  $N$  do
3:    $x_1^{(i)} \sim q_0(x)$ 
4:    $w_1^{(i)} = \frac{1}{N} k_0$ 
5: end for
6:  $n \leftarrow n + 1$ 
7: for  $i = 1$  to  $N^{(1)}$  do
8:    $P_{n,1}^{(i)} \sim q_n^{(1)}(p)$ 
9:    $X_{n-1,1}^{(i)} \sim q_n^{(1)}(x_{n-1} | p_{n,1}^{(i)})$ 
10:   $X_{n,1}^{(i)} \sim q_n^{(1)}(x_n | x_{n-1,1}^{(i)}, p_{n,1}^{(i)})$ 
11: end for
12: for  $i = 1$  to  $N^{(2)}$  do
13:   $X_{n-1,2}^{(i)} \sim q_n^{(2)}(x_{n-1})$ 
14:   $X_{n,2}^{(i)} \sim q_n^{(2)}(x_n | x_{n-1,2}^{(i)})$ 
15: end for
16: for  $p = 1$  to  $m_n$  do
17:   if  $N_p^{(1)} > 0$  then
18:     compute  $\tilde{Z}_{n,p}$  as per equation (15)
19:   end if
20: end for
21: for  $i = 1$  to  $N^{(1)}$  do
22:   compute  $W_{n,1}^{(i)} = w_n^{(1)}(X_{n,1}^{(i)}, X_{n-1,1}^{(i)}, P_{n,1}^{(i)})$  as per equation (16)
23: end for
24: for  $i = 1$  to  $N^{(2)}$  do
25:   compute  $W_{n,2}^{(i)} = w_n^{(2)}(X_{n,2}^{(i)}, X_{n-1,2}^{(i)})$  as per equation (17)
26: end for
27: obtain  $\{W_n^{(i)}, X_n^{(i)}\}_{i=1}^N$  by taking the union of  $\{W_{n,1}^{(i)}, X_{n,1}^{(i)}\}_{i=1}^{N_1}$  and  $\{W_{n,2}^{(i)}, X_{n,2}^{(i)}\}_{i=1}^{N_2}$ , and relabelling
28: the empirical measure  $\hat{\alpha}_n^N(dx_n) = \sum_{i=1}^N W_n^{(i)} \delta_{X_n^{(i)}}(dx_n)$  approximates  $\hat{\alpha}_n(dx_n)$ 
29: Goto 6

```

Fig. 1. Auxiliary SMC PHD algorithm

$$x_n = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{n-1} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

where v_1, v_3 and v_2, v_4 are i.i.d. zero mean Gaussian with standard deviation $\sigma_{x,1} = 0.02$ and v_2, v_4 are i.i.d. zero mean Gaussian with standard deviation $\sigma_{x,2} = 0.3$. Probability of survival is set $p_S = 0.98$ and the birth intensity is defined as $\gamma = 0.2\mathcal{N}(\cdot; x_b, \Sigma_b)$, where:

$$x_b = \begin{bmatrix} 30 \\ 0 \\ 30 \\ 0 \end{bmatrix}, \quad \Sigma_b = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

A. Example 1

In this first example, for illustrative purposes only, the position of the target is observed in additive, isotropic Gaussian noise with standard deviation $\sigma_z = 0.2$. Ground truth data is shown in Fig. (3). We explore two clutter scenarios. In both scenarios, the clutter intensity is set uniform on $[0, 100] \times [0, 100]$. In the first case $\kappa = 0.001$, corresponding to an average of 10 clutter points per scan, and in the second case $\kappa = 0.005$, corresponding to an average number of 50 clutter points per scan. In both cases we set $p_D = 0.99$.

While the structure of this model is simple, the low observation noise is a challenge for SMC algorithms. The localization of the likelihood means that blind proposals have little chance of putting particles in regions of high weight.

In the two clutter scenarios we tested the reliability of the proposed algorithm relative to the bootstrap PHD filter, by repeatedly running the algorithm over fixed observation records generated from the model. In both cases the GM-PHD filter of [27] was used to compute a ‘baseline’ result. This essentially allows the performance of the SMC algorithms to be tested against a very close approximation of the true PHD filter for this model. For the GM-PHD algorithm, the total number of mixture components was set to 100, the merging radius was set to 4 and the pruning threshold was set to 10^{-5} . It was found that using more conservative values did not significantly change the performance of the algorithm.

In the two scenarios, due to the linear-Gaussian nature of the model, it is possible to employ the optimal proposal distributions described in section V, where in the case of $q_n^{(1),opt}(p)$, $\hat{\alpha}_{n-1}^N$ used in place of $\hat{\alpha}_{n-1}$. It is also possible to employ the optimal particle allocation of Corollary 1, with $N^{(2)}$ rounded up to the nearest integer. Again $\hat{\alpha}_{n-1}^N$ was used in place of $\hat{\alpha}_{n-1}$ for computation of $N^{(1)}$ and $N^{(2)}$.

1) *Low Clutter*: In the low clutter scenario, the two SMC algorithms were implemented with $N = 500$ particles, with 170 allocated to births for the bootstrap algorithm. Both algorithms were initialized with $k_0 = 2$, sampling from $\mathcal{N}(\cdot; x_b, \Sigma_b)$.

In order to demonstrate the superior performance of the proposed algorithm we focus on two characteristics of the particle set: its total mass at each iteration and the Effective Sample Size - a standard quantity for assessing the efficiency of SMC algorithms (further details given below).

We first discuss results for the low clutter scenario. Fig. (2) shows images of normalized histograms of the total mass of the particle set for the bootstrap and auxiliary algorithms. The observation record was of length $n = 40$. For each image, a vertical strip corresponding to a single time index shows a normalized histogram of the total mass of the particle set, over 100 independent runs of the algorithm. The darker the pixel, the more frequently the total mass took the corresponding value. Super-imposed is the true number of targets (dashed line) and the total mass of the PHD computed using the GM-PHD filter (solid line) pre-pruning/merging. It should also be noted that the total mass of the the particle set is independent of whatever heuristic clustering device is used to extract target state estimates. Thus we concentrate on the efficiency of the SMC techniques.

There are two important features to note. Firstly, the image for the bootstrap algorithm indicates greater variation in the output of the algorithm across independent runs, compared to the auxiliary algorithm. Secondly, the bootstrap algorithm frequently under-estimates the number of targets between time steps 20 and 40 as during these iterations the total mass of the particle set is very frequently lower than the true number of targets present. This is due to the bootstrap algorithm losing track of targets and failing to identify the birth of a target at $n = 11$. By contrast, the auxiliary algorithm is able to track reliably all the targets. It also more closely follows the total mass computed using the GM-PHD filter.

The total mass of the GM-PHD filter is generally slightly higher than that obtained from the SMC algorithms. We conjecture that in the case of this model the GM-PHD filter is able to accurately represent the large number of small modes in the true PHD. The SMC methods often neglect these smaller modes, with particles concentrated on the more significant modes. We further conjecture that the peaks in the total mass of the intensity can be explained as follows. Any observation which is made close to the birth region contributes positive mass to the updated intensity, irrespective of whether or not it originates from a hidden target. As pointed out by one anonymous reviewer, using a lower value of Γ might reduce this effect.

We note that it is recommended in [27] that the mixture components in the GM-PHD filter should be thresholded according to their weight post pruning/merging, in order to estimate the number of targets present. We have chosen to show the total mass pre-thresholding, as the aim of this example is to show how well the proposed algorithm approximates the true PHD.

As noted in section II, the peaks in the intensity function can be used to obtain state estimates for individual targets. In existing particle implementations of the PHD filter, this requires heuristic clustering of the particles, followed by estimation within each cluster. For the auxiliary particle PHD filter, there is a natural (albeit still heuristic) method which can be employed to this end without the increased computational cost of employing a clustering algorithm. This consists of simply computing estimates from particles clustered by common values of $P_n^{(i)}$. An estimate is taken when the total weight of the cluster exceeds some threshold, e.g. 0.5. This method can be expected to work well when the clutter intensity is not too high. For each observation with a positive number of particles assigned to

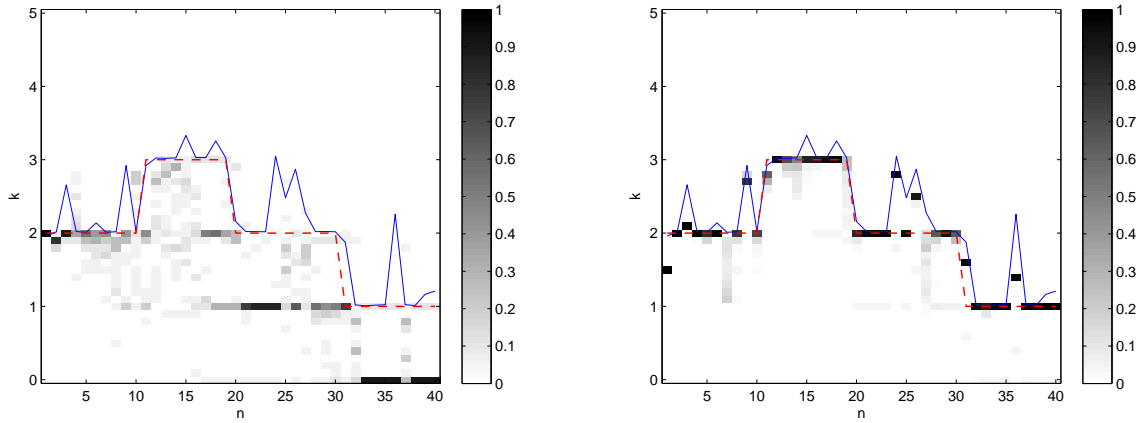


Fig. 2. Low clutter scenario. Normalized histograms of the total mass of the particle set. Each vertical strip in the image corresponds to a normalized histogram of the total mass of the particle set at that time index, averaged over 100 independent runs of the algorithm. Left: Bootstrap filter. Right: Auxiliary filter. Superimposed plots in both cases show the true number of targets (dashed) and the total mass of the updated intensity computed by the GM-PHD algorithm (solid).

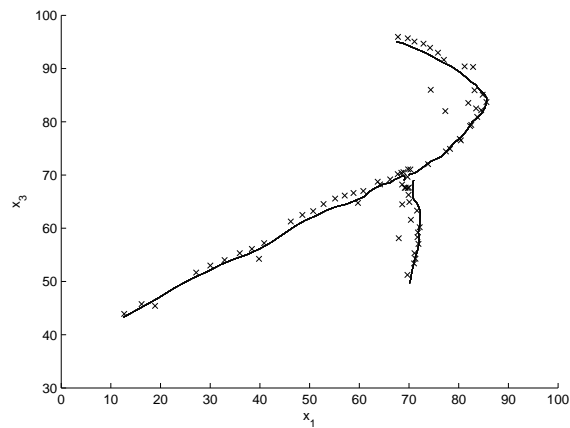


Fig. 3. Low clutter scenario. Ground truth target positions (solid line) and state estimates (crosses) for auxiliary filter. 10 clutter points per scan.

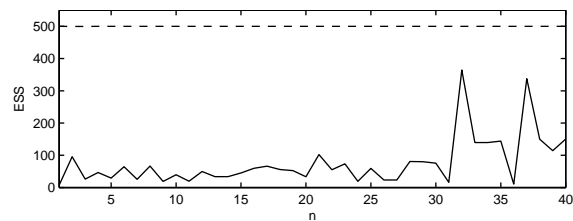


Fig. 4. Low clutter scenario. Effective sample size versus iteration of the algorithm, averaged over 100 runs with $N = 500$ particles. Dashed: Auxiliary filter. Solid: Bootstrap filter. 10 clutter points per scan.

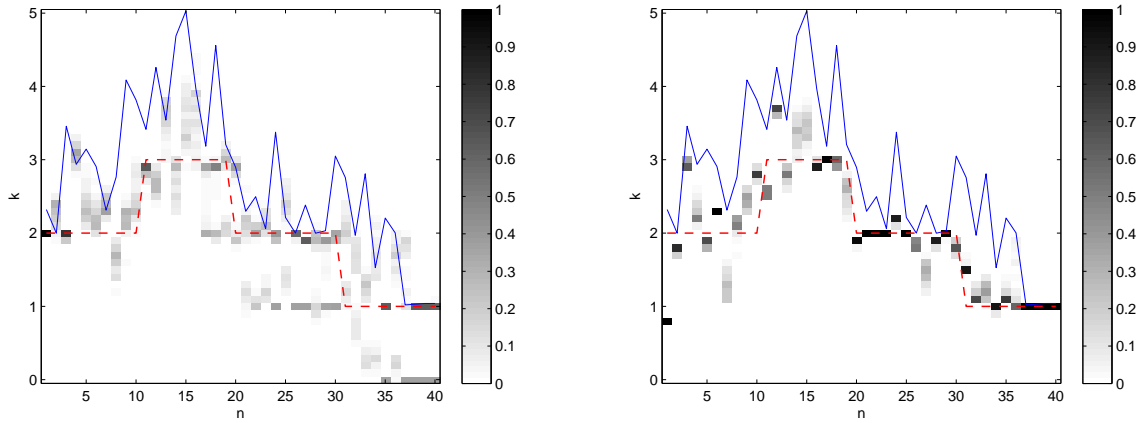


Fig. 5. High clutter scenario. Normalized histograms of the total mass of the particle set. Each vertical strip in the image corresponds to a normalized histogram of the total mass of the particle set at that time index, averaged over 100 independent runs of the algorithm. Left: Bootstrap filter. Right: Auxiliary filter. Superimposed plots in both cases show the true number of targets (dashed) and the total mass of the updated intensity computed by the GM-PHD filter (solid).

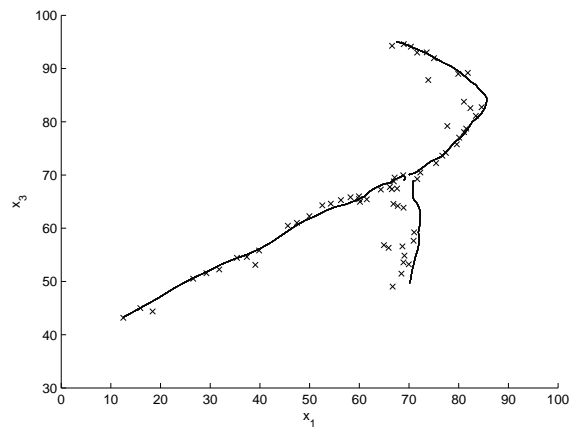


Fig. 6. High clutter scenario. Ground truth target positions (solid line) and state estimates (crosses) for auxiliary filter. 50 clutter points per scan.

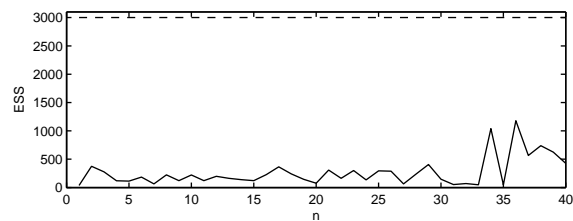


Fig. 7. High clutter scenario. Effective sample size versus iteration of the algorithm, averaged over 100 runs with $N = 3000$ particles. Dashed: Auxiliary filter. Solid: Bootstrap filter. 50 clutter points per scan.

it, a state estimate can be obtained by normalizing the weights of the corresponding particles and taking the weighted mean of the particle locations. This method for state estimation bears some similarities to that proposed for the GM-PHD filter in [27]: in the latter case, state estimates are drawn from individual Gaussian mixture components (after thresholding by mass), each of which is associated with an observation index $p \in \{1, \dots, m_n\}$.

Estimates computed using the proposed clustering method for a single run of the auxiliary particle algorithm in the low clutter scenario are shown in Fig. (3). The solid lines in this plot show the true position trajectories from which the observations were generated.

Secondly, the effective sample size (ESS), introduced in [33], provides a measure of the degeneracy of the importance weights. It has a maximum value equal to the total number of particles. In practice, the ESS may not be evaluated exactly but it can be estimated from the particle set as follows:

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\hat{w}_n^{(i)})^2},$$

where $\{\hat{w}_n^{(i)}\}_{i=1}^N$ are the normalised importance weights. When the ESS is small, SMC algorithms can collapse giving very high variance estimators.

Fig. (7) shows the ESS of the normalized particle sets, calculated at each iteration and then averaged across the 100 runs. As the auxiliary algorithm uses the optimal proposal distributions the variation of the importance weights is tiny (in fact the only variation occurs because $N^{(2)}$ is rounded up to the nearest integer). At all iterations, the average ESS for the bootstrap algorithm is lower than that for the auxiliary algorithm.

In terms of computational cost, the proposed algorithm was found to be similar to the bootstrap algorithm. In general, the auxiliary approach incurs costs associated with computing the parameters of proposal distributions which the bootstrap algorithm does not. However, this is balanced by the inexpensive state-estimation scheme which can be employed with the auxiliary algorithm. Both SMC methods are typically more expensive than the GM-PHD algorithm, which does not involve random number generation.

2) *High Clutter*: In the high clutter scenario both algorithms had $N = 3000$ of which 1000 were allocated to births for the bootstrap algorithm. Both algorithms were initialized in the same manner as in the low clutter case.

We now discuss results in the high clutter scenario. The images of Fig. (5) again indicate that the proposed algorithm is more reliable than the bootstrap algorithm: across 100 independent runs the total mass of the particle set exhibits less variation so the pixels in the image plot for the auxiliary algorithm are less ‘smeared-out’. As in the low clutter scenario, the bootstrap algorithm frequently loses track of one target and so the total mass of the particle set is often lower than the true number of targets between time indices 20 and 40. In the high clutter scenario, the GM-PHD filter again accurately portrays many small modes in the true PHD so its total mass is again higher than that of the SMC algorithms. In figure Fig. (7), the ESS is again significantly higher for the auxiliary algorithm compared to the bootstrap method. Position estimates obtained from the auxiliary algorithm using the method described above are shown in Fig. 6.

B. Example 2

In this second example, we use the same dynamical model and true data as in Example 1, but with a range and bearing observation model:

$$\begin{aligned} y_n^{(r)} &= \sqrt{x_{n,1}^2 + x_{n,3}^2} + u_{n,r}, \\ y_n^{(b)} &= \arctan\left(\frac{x_{n,3}}{x_{n,1}}\right) + u_{n,b}, \end{aligned}$$

where $u_{n,r}$ and $u_{n,b}$ are zero mean Gaussian disturbances of variance σ_r^2 and σ_b^2 respectively.

For this challenging nonlinear observation model, we demonstrate the performance of the proposed algorithm using the UT to approximate the optimal proposal distributions. The UT employs a set of deterministic weighted points (called *sigma points*) to capture the mean and covariance of a probability distribution. A Kalman recursion can then be executed on these quantities yielding a deterministic approximation to the optimal filter, see [46], [12], [47] for details.

We use the UT to approximate each $V_{n,p}^{(1),\text{opt}}$ in the same manner that it is used to approximate the predictive likelihood in [45]. This requires sigma points and associated weights to be calculated for each particle and the mean of the normalized birth intensity, at each iteration. For each p , $q_n(p)$ and $q_n^{(1)}(dx_{n-1}|p)$ are constructed in terms of:

$$V_{n,p}^{(1)}(x_{n-1}^{(i)}) = \sum_{j=1}^{\nu} \psi_{n,p}(\zeta_{n-1,j}^{(i)}) \lambda_{n-1,j}^{(i)} \quad (20)$$

where $\{\zeta_{n-1,j}^{(i)}\}_{j=1}^{\nu}$ and $\{\lambda_{n-1,j}^{(i)}\}_{j=1}^{\nu}$ are respectively the locations and weights of the sigma points which collectively capture the mean and covariance of $f(x_n|x_{n-1}^{(i)})$ for the i th particle in $\alpha_{n-1}^N(x_{n-1})$. The same procedure is also carried out once at each iteration for the birth term to obtain $V_{n,p}^{(1)}(s)$ in terms of sigma points which capture the characteristics of $\gamma(x)/\Gamma$.

The value of ν is dictated by the UT. The scheme in [46] has $\nu = 2d + 1$ where d is the dimension of E . In the case of this example, $\nu = 9$, which is dominated in order of magnitude by the total number of particles employed. Therefore the arithmetic operations involved in computing the sigma points, the weights and (20) for each particle do not significantly increase the overall computational expense of the algorithm.

Finally, the proposal distribution $q_n^{(1)}(dx_n|x_{n-1}^{(i)}, p)$ is obtained by passing the mean and covariance captured by the sigma points of $x_{n-1}^{(i)}$ (the same point locations and weights which have already been computed) through the update operation of the Kalman filtering recursion, using $\psi_{n,p}(x_n)$ as the likelihood function, see [12] for details. This yields a Gaussian approximation to $q_n^{(1),\text{opt}}(dx_n|x_{n-1}^{(i)}, p)$.

In this example we use the same dynamical noise variances and the same birth intensity as in example 1, (19). The clutter intensity is set to be uniform over $[0, 141] \times [0, \pi/2]$, with an average of 20 clutter points per scan. The standard deviation of the bearing noise was set to $\sigma_b = 0.002$. We consider various values of the σ_r , the standard deviation of the range observation noise and we set $p_D = 0.98$ and $p_S = 0.98$.

We compare the proposed algorithm to the bootstrap SMC PHD filter and the UKF-type Gaussian Mixture (UKF-GM) implementation of the PHD filter of [27]. For the latter we set the maximum number of mixture components post pruning/merging to 100, the merge radius to 4 and the pruning threshold to 10^{-5} . We also use the scheme advocated in [27] to extract point estimates: after pruning/merging, the means of mixture components with mass greater than 0.5 were taken as point position estimates. We note using different values for any of these parameters did not significantly improve the performance of the algorithm.

Both SMC algorithms used a total of 1500 particles, with 500 being assigned to births in the bootstrap algorithm. For the auxiliary algorithm we fixed $N^{(2)} = 150$.

In order to compare the quality of the position estimates from the UKF-GM algorithms with those obtained from the proposed algorithm, we computed the Wasserstein Distance (WD) between the set of point position estimates and the true target positions at each iteration (see [48] for further information about this metric). For the proposed algorithm, the natural clustering mechanism for obtaining state estimates was employed, as described above.

Comparisons were made by computing the average WD over the 40 iterations of the algorithm, for various values of the range noise standard deviation σ_r . Results are presented in figure 8. When σ_r was small, it was found that the UKF-GM filter worked well, but as σ_r was increased, its performance degraded.

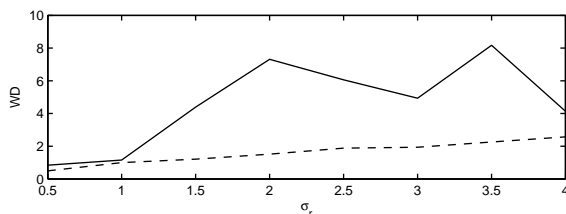


Fig. 8. Wasserstein distance between position estimates and true values versus range observation noise standard deviation, averaged over 40 iterations of algorithm. Dashed: Auxiliary filter. Solid: UKF-GM filter.

In order to better compare the reliability of the proposed algorithm to the UKF-GM and bootstrap algorithm, the total mass of the corresponding approximations of the total mass of the PHD were recorded over 100 independent runs of the algorithm, each with a different observation record and with fixed $\sigma_r = 1.5$. This allows us to portray the robustness of the filtering algorithms to variation in observation records. Fig. (9) shows normalized histograms of the total mass of the PHD from the three algorithms in the same format as the image plots in Example 1. Again, a dark pixel indicates that the total mass of the PHD frequently took that value across runs of the algorithm. The dashed line indicates the true number of targets. For this example, we are unable to compute the true PHD, due to the non-linearity of the model.

The results for the bootstrap algorithm indicate that it is somewhat unreliable, with few dark pixels in the image plots. As in Example 1, the total mass of the particle set is frequently less than the true number of targets between time indices 20 and 40. For the UKF-GM filter, the total mass fluctuates widely across different observation records. It seems reliable over early iterations, where the total mass frequently matches the true number of targets, but at

subsequent iterations the total mass is often significantly lower than the true number of targets and at the later iterations it appears to sometimes ‘lose-track’ altogether. For the auxiliary algorithm, the dark pixels indicate that the total mass is frequently close to the true number of targets, although at later iterations the mass is occasionally lower.

We note that there are fewer spikes in the total mass of the particle set compared to Example 1. However, in the image plots of Example 1 it should be noted that we considered a single fixed observation sequence, where as in this example we consider results obtained over different observation records generated from the model. Hence the small spikes in the PHD are averaged out and do not feature significantly in the image plots.

For further comparison of the proposed method with the bootstrap SMC algorithm we again compute the ESS and average over 100 runs of the algorithm. The results are shown in Fig. 10. The ESS is again higher for the proposed method than for the bootstrap algorithm. The only significant transient features in the the ESS for the proposed algorithm are after initialisation and around $n = 11$ when the third target appears.

Finally, position estimates obtained from the proposed algorithm are plotted in figure 11 along with the true position trajectories. The same method for state extraction was used as described in Example 1.

VII. CONCLUSIONS

We have introduced an auxiliary particle implementation of the PHD filter. The proposed scheme involves auxiliary random variables which index observations and pre-selection of particles on a per-observation basis in a manner similar to the APF. The resulting algorithm samples on a higher dimensional space than previous particle implementations of the PHD filter, but doing so permits more efficient proposal mechanisms.

In SMC algorithms, it is importance to minimise the variance of the importance weights in order to obtain low-variance estimators. In existing implementations of the PHD filter it is not clear how to do so. We have provided guidance on choices of proposal distributions which are optimal in this sense and provided interpretation of them in terms of the PP model underlying the PHD recursion. Specifically, the value of the optimal proposal distribution probability for an observation index has been shown to be proportional to the posterior probability that the corresponding observation originates from a true target, given a Poisson prior. The proposed method exhibits a natural mechanism for clustering of particles on the basis of the observations to which they are assigned. This can be used as a computationally inexpensive tool for extracting state estimates. Numerical results have demonstrated the gains in efficiency which are possible using the proposed approach.

As pointed out by one anonymous reviewer, it may be possible to apply similar techniques for implementation of the CPHD filter. This is a possible avenue for future work.

REFERENCES

- [1] R. P. S. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1152–1178, October 2003.
- [2] —, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

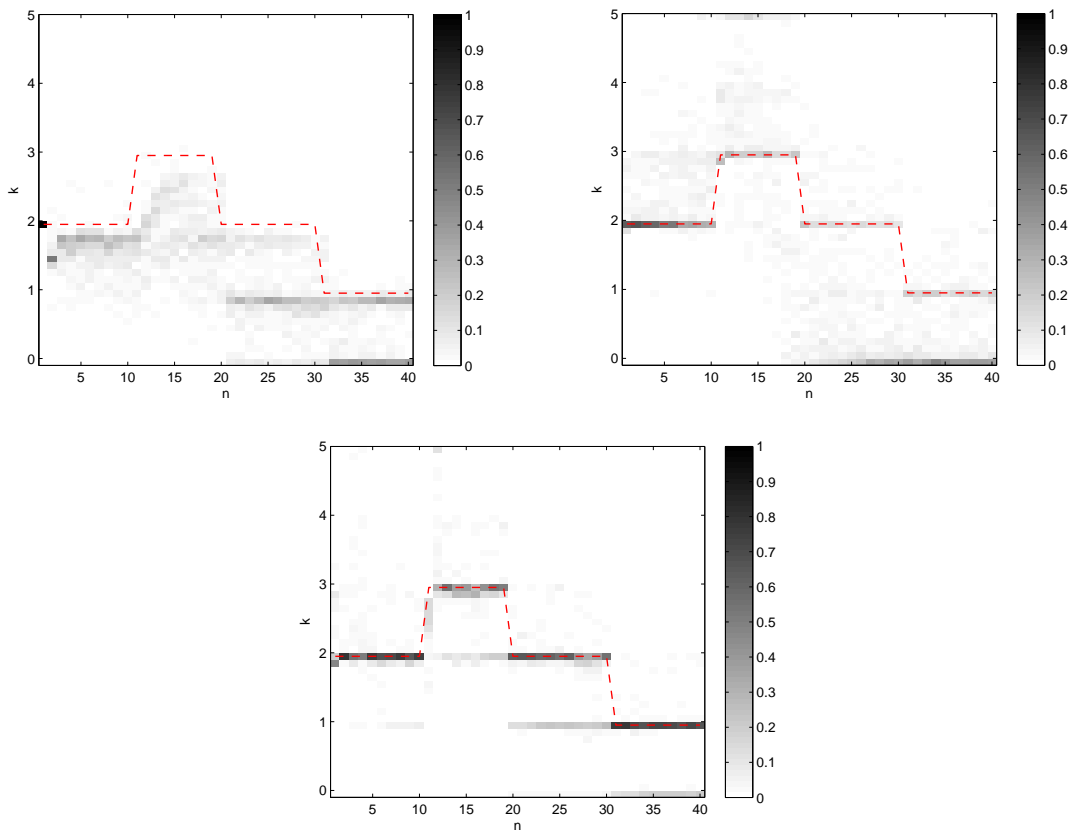


Fig. 9. Normalized histograms of the total mass of the particle set. Each vertical strip in the image corresponds to a normalized histogram of the total mass of the particle set at that time index, averaged over 100 independent runs of the algorithm, each with a different observation record. Top left: Bootstrap filter. Top right: UKF-GM filter. Bottom: Auxiliary filter. Superimposed plot (dashed) is the true number of targets.

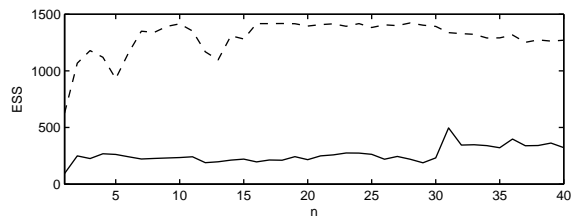


Fig. 10. Effective Sample size vs iteration of the algorithm, averaged over 100 runs, each with a different observation record. Dashed: Auxiliary filter. Solid: Bootstrap filter.

- [3] S. Singh, B.-N. Vo., A. Baddeley, and S. Zuyev, “Filters for spatial point processes,” University of Cambridge, Department of Engineering – Signal Processing Lab, Tech. Rep. CUED/F-INFENG/TR.591, 2007.
- [4] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.
- [5] R. Mahler, “Phd filters of higher order in target number,” *IEEE Trans. on Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, October 2007.
- [6] E. Biglieri and M. Lops, “Multiuser detection in a dynamic environment - part I: User identification and data detection,” *IEEE Trans. Inf.*

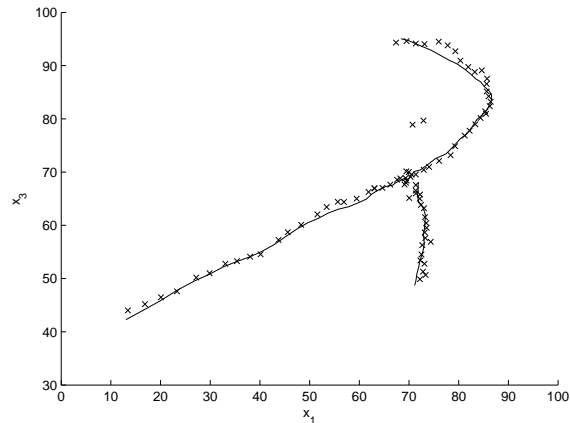


Fig. 11. Ground truth target positions (solid line) and state estimates (crosses) for auxiliary PHD filter.

Theory, vol. 53, no. 9, pp. 3158–3170, September 2007.

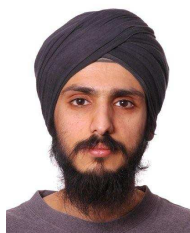
- [7] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, “Particle PHD filter for multi-target visual tracking,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, Hawaii.*, vol. 1, 2007, pp. 1101–1104.
- [8] W. K. Ma, B. Vo, S. Singh, and A. Baddeley, “Tracking an unknown and time varying number of speakers using TDOA measurements: A random finite set approach,” *IEEE Trans. Signal Processing*, vol. 54, no. 9, pp. 3291–3304, 2006.
- [9] D. Clark, A. T. Cemgil, P. Peeling, and S. Godsill, “Multi-object tracking of sinusoidal components in audio with the Gaussian mixture probability hypothesis density filter,” in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2007, pp. 339–342.
- [10] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. New York: Springer Verlag, 2001.
- [11] H. W. Sorenson and A. R. Stubberud, “Nonlinear filtering by approximation of the a posteriori density,” *International Journal of Control*, vol. 8, pp. 33–51, 1968.
- [12] S. Julier and J. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Proceedings of SPIE*, vol. 3068, 1997, pp. 182–193.
- [13] C. Hue, J.-P. L. Cadre, and P. Perez, “Tracking multiple objects with particle filtering,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 791–812, July 2002.
- [14] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy, “Particle filtering for multi-target tracking and sensor management,” in *Proceedings of the Fifth International Conference on Information Fusion*, vol. 1, 2002, pp. 474–481.
- [15] J. Vermaak, S. Godsill, and P. Perez, “Monte Carlo filtering for multi target tracking and data association,” *IEEE Transactions Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, January 2005.
- [16] W. Ng, J. Li, S. Godsill, and S. K. Pang, “Multitarget initiation, tracking and termination using Bayesian Monte Carlo methods,” *The Computer Journal*, vol. 50, no. 6, pp. 674–693, 2007.
- [17] M. R. Morelande, C. Kreucher, and K. Kastella, “A Bayesian approach to multiple target detection and tracking,” *IEEE Trans. Signal Processing*, vol. 55, no. 5, pp. 1589–1604, May 2007.
- [18] M. Vihola, “Rao-blackwellised particle filtering in random set multitarget tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 689–705, April 2007.
- [19] B. Vo, S. Singh, and A. Doucet, “Random finite sets and sequential Monte Carlo methods in multi-target tracking,” in *Proceedings of the International Conference on Information Fusion, Cairns, Australia*, 2003, pp. 792–799.
- [20] H. Siddenblath, “Multi-target particle filtering for the probability hypothesis density,” in *Proceedings of the International Conference on Information Fusion, Cairns, Australia*, 2003, pp. 800–806.

- [21] T. Zajic and R. P. S. Mahler, "Particle-systems implementation of the PHD multitarget tracking filter," in *Proceedings of SPIE*, 2003, pp. 291–299.
- [22] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.
- [23] N. Whiteley, S. Singh, and S. Godsill, "Auxiliary particle implementation of the probability hypothesis density filter," in *Proceedings of IEEE 5th International Symposium on Image and Signal Processing and Analysis (ISPA)*. Istanbul, Turkey., September 2007.
- [24] B. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. on Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, October 2005.
- [25] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*, 2nd ed., ser. Probability and Its Applications. New York: Springer, 2003, vol. I: Elementary Theory and Methods.
- [26] J. F. C. Kingman, *Poisson Processes*, ser. Oxford Studies in Probability. Oxford University Press, 1993.
- [27] B. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4091–4104, November 2006.
- [28] D. Clark and B. Vo, "Convergence analysis of the gaussian mixture PHD filter," *IEEE Trans. Signal Processing*, vol. 55, no. 4, pp. 1204–1212, April 2007.
- [29] B. Vo, B. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Processing*, 2007, to appear.
- [30] O. Cappé, S. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 899–924, May 2007.
- [31] P. Del Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo methods for Bayesian computation," in *Bayesian Statistics 8*. OUP, 2006.
- [32] C. Robert and G. Casella, *Monte Carlo Statistical Methods*, ser. Springer Texts in Statistics. Springer, 2004.
- [33] 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.
- [34] N. Chopin, "Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference," *Annals of Statistics*, vol. 32, no. 6, pp. 2385–2411, 2004.
- [35] P. Del Moral, *Feynman-Kac formulae: genealogical and interacting particle systems with applications*, ser. Probability and Its Applications. New York: Springer Verlag, 2004.
- [36] 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.
- [37] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.
- [38] S. Godsill and T. Clapp, "Improvement strategies for Monte Carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science, A. Doucet, J. F. G. D. Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001, ch. 7.
- [39] A. M. Johansen and A. Doucet, "A note on the auxiliary particle filter," *Statistics and Probability Letters*, 2008, to appear.
- [40] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularized particle filters," in *Sequential Monte Carlo Methods In Practice*, ser. Statistics for Engineering and Information Science, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer Verlag, 2001, ch. 13.
- [41] A. M. Johansen, S. Singh, A. Doucet, and B. Vo, "Convergence of the SMC implementation of the PHD filter," *Methodology and Computing in Applied Probability*, vol. 8, no. 2, pp. 265–291, June 2006.
- [42] D. E. Clark and J. Bell, "Convergence results for the particle PHD filter," *IEEE Trans. Signal Processing*, vol. 54, no. 7, pp. 2652–2661, July 2006.
- [43] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, September 1998.
- [44] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering," in *Proceedings of the IEEE 4th International Symposium on Image and Signal Analysis*, 2005, pp. 64–69.

- [45] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1762–1770, July 2003.
- [46] S. Julier and J. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," University of Oxford, Department of Engineering, Tech. Rep., 1996.
- [47] S. Julier, "Reduced sigma points filters for the propagation of means and covariances through nonlinear transformations," in *Proceedings of American Control Conference*, vol. 2, 2002, pp. 887–892.
- [48] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Syst. Man Cybernetics*, vol. 43, no. 3, pp. 327–336, 2004.



Nick Whiteley received the M.Eng. degree from Oxford University in 2005. From October 2005 to September 2008 he studied for the Ph.D. degree in the Signal Processing Laboratory, Department of Engineering, at the University of Cambridge. Since October 2008 he has been a Brunel Post-Doctoral Research Fellow in the Statistics Group, Department of Mathematics at the University of Bristol. His research interests lie in Monte Carlo methods for computation in statistical inference and optimization problems



Sumeetpal Singh was awarded the Ph.D. degree in 2002 in Electrical Engineering from the University of Melbourne. Form 1997 to 1999 he worked as a wireless communications Design Engineer. From 2003 to 2004 he was a Research Associate at Melbourne University and joined the Cambridge University Engineering Department as a Research Associate in March 2004. In March 2007 he was appointed as a University Lecturer in Engineering Statistics. He is also a Fellow of Churchill College and an Affiliated Lecturer at the Statistics Laboratory of Cambridge University. His research focus is on Statistical Signal Processing, in particular, Monte Carlo methods for estimation and control.



Simon Godsill is Professor of Statistical Signal Processing in the Engineering Department of Cambridge University. He has acted as an Associate Editor for *IEEE Trans. Signal Processing* and the journal *Bayesian Analysis*, and as a member of *IEEE Signal Processing Theory and Methods* Committee. He has research interests in Bayesian and statistical methods for signal processing, Monte Carlo algorithms for Bayesian problems, modelling and enhancement of audio and musical signals, tracking and high frequency financial data. He has published extensively in journals, books and conferences. He has co-edited in 2002 a special issue of *IEEE Trans. Signal Processing* on Monte Carlo Methods in Signal Processing and organised many conference sessions on related themes. He is currently co-organising a year long program on Sequential Monte Carlo Methods at the SAMSI Institute in North Carolina.