

Two-layer particle filter for multiple target detection and tracking

Ángel F. García-Fernández, Jesús Grajal, Mark R. Morelande

Abstract

This paper deals with the detection and tracking of an unknown number of targets using a Bayesian hierarchical model with target labels. To approximate the posterior probability density function, we develop a two-layer particle filter. One deals with track initiation, and the other with track maintenance. In addition, the parallel partition method is proposed to sample the states of the surviving targets.

Index Terms

Two-layer particle filter, parallel partition method, detection, tracking, target labels.

NOMENCLATURE

\mathbf{P}^k	Vector that contains the positions of all tracks at time k
$\mathbf{p}_{j,i}^k$	i th position particle of track j at time k
\mathbf{p}_j^k	Position vector of track j at time k
\mathbf{R}^k	Augmented multitrack state vector j at time k
\mathbf{r}_j^k	Augmented state vector of track j at time k
$\mathbf{X}^{k+1}(S^k)$	Multitrack state vector of the targets whose labels are in set S^k at time $k + 1$

Copyright (c) 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Ángel F. García-Fernández was supported by an FPU Fellowship from Spanish MEC. This work was supported in part by the Spanish National Research and Development Program under Projects TEC2011-28683-C02-01 and Comonsens (Consolider-Ingenuo 2010, CSD2008-00010). The authors would like to thank Wei Yi for helpful comments.

*Ángel F. García-Fernández and Jesús Grajal are with the Departamento de Señales, Sistemas y Radiocomunicaciones, ETSI de Telecomunicación, Universidad Politécnica de Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain. email: {agarcia, jesus}@gmr.ssr.upm.es. Mark R. Morelande is with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia. email: mrmore@unimelb.edu.au.

$\mathbf{x}^{k+1}(s_j^k)$	State vector of the target whose label is s_j^k at time $k + 1$
\mathbf{X}^k	Multitrack state vector at time k
\mathbf{x}_j^k	State vector of track j at time k
$\mathbf{x}_{j,i}^k$	Rao-Blackwellised i th particle of track j at time k
$\mathbf{z}^{1:k}$	Sequence of measurement vectors from time 1 to k
\mathbf{z}^k	Measurement vector at time k
b^k	Number of targets that are born at time k
d^k	Number of targets that die at time k
I^k	Set of labels at time k
N^k	Set of labels of the targets that are born at time k
S^k	Set of labels of the targets that survive from time k to $k + 1$
t^k	Number of tracks at time k

I. INTRODUCTION

The random finite set (RFS) framework has emerged as a natural way to perform multiple target detection and tracking from the Bayesian point of view [1]. In the RFS framework, the multitarget state is a set whose elements are single target state vectors. Densities within the RFS framework are manipulated using rules analogous to those which are applied in vector-based approaches [2]. Thus, the (unnormalised) RFS posterior density can be found using an RFS analog of the Chapman-Kolmogorov equation followed by multiplication of the result by an RFS version of the likelihood. Importantly, this framework permits the development of rigorous metrics for algorithm performance assessment [3], [4]. In the usual case of RFS, the state vectors do not include identifying labels and, therefore, we estimate the target number and target states without connecting these estimates, instant to instant, into temporally connected tracks [5].

The joint multitarget probability density (JMPD) [6]–[8] is another framework to estimate the target number and target states without building temporally connected tracks. The JMPD is a vector-based approach in which the quantity of interest is a hybrid state formed by the target number and a multitarget state, which is a vector formed by concatenating the single target state vectors. This hybrid state does not include identifying labels. This results in the impossibility of the JMPD to build temporally connected tracks. In [6]–[8], the (unnormalised) JMPD is found by the usual Bayesian filtering recursion involving evaluation of the Chapman-Kolmogorov equation followed by the multiplication of the result by the likelihood. This simplifies the calculation of the posterior in comparison with the RFS framework mainly due to the use of vector integrals rather than set integrals. However, the JMPD approach is only sound if

the posterior remains unchanged under target permutations [9, Appendix B]. This leads to a theoretical inconsistency since the Bayesian filtering recursion for vector-valued variables does not generally provide the required symmetry [10]. Rather, this symmetry must be imposed. The resulting JMPD actually ends up being equivalent to the RFS posterior but is not satisfying theoretically. As such, the RFS framework should be used if individual target states are not labelled.

The essential task of linking target state estimates along time to form tracks can be done if a unique label is added to each single target state in both approaches. Labels differ from the usual kinematic parameters in two important ways: they are unique (no two tracks can have the same label) and they are fixed in time. Track construction through labelling has been proposed in several papers, although usually with *ad hoc* approaches. In [7] labels are added in the particle filter (PF) approximation to the JMPD but they are not included in the Bayesian model. The same happens in the PF approximation to the probability hypothesis density (PHD) filter, which is an approximation to the full RFS posterior, proposed in [11]. As labels are included in the target state, a principled approach must take them into account directly in the Bayesian model not in an approximation to a theoretically unlabelled density. An example of this principled approach to track labelling within the RFS framework can be found in [12]. It is important to note that the two distinguishing characteristics of the labels imply some important changes with respect to the unlabelled target case. As pointed out in [9, Appendix B], now there is a bijection between the multitarget RFS and a hybrid state formed by the target number and a multitarget state vector with labels. Therefore, because of this one-to-one correspondence, both representations lead to the same results and, importantly, they are equally sound. Additionally, this one-to-one correspondence implies that any metric defined in an RFS framework can be equivalently defined in this framework. Since a vector-based approach with labels conveys the same information as an RFS approach with labels we will pursue the former approach because some parts of this paper are clearer this way. Nevertheless, this paper could have been written using RFS notation as they are equivalent. We note that hierarchical models of the type used here have been previously used in multitarget tracking in [13] and, more generally, are commonly used in Bayesian inference [14]–[16].

This paper is concerned with PF approximations to the posterior PDF using the above-mentioned hierarchical model with labels. In this context, timely and reliable detection of targets and accurate target state estimation require an importance density that efficiently samples both target states and the number of targets. Although careful design of the importance density is necessary for any importance sampling approximation, the potentially high dimension of the multitarget state, which is composed of all the individual target states, make this especially crucial in multitarget tracking [17]. The following

paragraphs discuss existing work on these problems and summarise how the proposed methods address the limitations of these approaches.

An important property of the measurement model commonly used in multitarget tracking is the approximate factorisation of the likelihood. This factorisation depends on the measurement model but usually applies for widely separated targets. As a result of this property only the states of targets which form clusters based on proximity need to be sampled jointly. This effectively reduces the dimensionality of the sampling problem. This idea was first exploited for sequential importance sampling in [18] and also used in [6]–[8], [19]–[21]. Although distant targets can be ignored it is desirable that samples of targets within a cluster be sampled jointly. This was achieved in [7] by jointly sampling clusters of target states from the optimal importance density (OID) [22]. The OID is not universally applicable because it is generally intractable although approximations to the OID using a bank of extended or unscented Kalman filters have been proposed [22], [23]. However, jointly sampling from the OID, or an approximation, can be computationally expensive, even for relatively small clusters of targets. The aim is therefore to construct an importance density which considers the presence of neighbouring targets in a computationally efficient manner. The proposed parallel partition (PP) method achieves this aim by a simple modification of the independent partition (IP) method of [6]. PP is a general method in the sense that it does not rely on a specific measurement model that is intended to improve the efficiency of sampling of the target states, especially when the number of particles is low and a fast algorithm is required under the assumption that tracks are a priori independent. In addition, we show that what the well-known IP method really does is sampling an auxiliary vector, therefore, providing more insight into multiple target tracking using PFs.

Efficient sampling of the target number is also vital in multitarget tracking. The simplest possibility is to draw the target number from an assumed prior [24]. This is highly inefficient as it removes and adds targets regardless of what the measurements suggest is appropriate. The notion of using a secondary approximation to the posterior to construct an importance density for sampling target number was used in [7], [25]. There, an existence grid was used to determine both regions likely to contain new targets and existing targets which are likely to be false. This approach has the advantage of using the available measurements rather than relying on a prior but does not incorporate target dynamics and is therefore inappropriate for even moderately rapid targets. In this paper a PF approximation, referred to as the pre-track filter, is used to construct an importance density for the addition of new tracks. Sampling of target number is performed via likelihood ratio tests (LRTs) on the pre-tracks and existing tracks. LRTs are commonly used for track scoring in the well known multiple hypothesis tracking (MHT) [26]. The LRT for the pre-tracks governs the addition of new targets, effectively transforming a pre-track to an

existing track if appropriate, while the LRT for the existing tracks determines target removal. Then, the PF developed in this paper has a two-layer scheme in which the posterior density of one layer is used to specify the importance density of the other layer to sample newborn target states. A similar idea, although not in the context of importance sampling, is used in the integrated probabilistic data association filter (IPDAF) and its generalisations where a target existence probability is used to determine the track status [27]. Nevertheless, the idea of using an subsidiary PF to sample the newborn target states is a new idea developed in this paper.

The rest of the paper is organised as follows. In Section II, we address the joint multitrack probability density (JMKPD) and describe the models of targets and measurements. In Section III, we explain the PP method for fixed and known number of targets. The two-layer PF for variable and unknown number of targets is presented in Section IV. Numerical examples examining the filter performance are provided in Section V. Finally, conclusions are drawn in Section VI.

II. JOINT MULTITRACK PROBABILITY DENSITY

This section deals with the JMKPD [28], the target model and the sensor model. Basically, the JMKPD corresponds to the approach used in [13] but augmenting each target state with a unique label so that the tracks are perfectly identified along time. Adding the unique labels to each track enables us to use an equivalent representation in which the dimension of the state is governed by the number of elements of a set that includes the labels so that the required PDFs are written in a more compact form. As noted in the Introduction, and shown in [9, Appendix B], the JMKPD is equivalent to the RFS posterior density of the multitarget state with labels.

We assume there are t^k tracks at time k . The state of track j at time k is described by vector $\mathbf{x}_j^k = [x_j^k, \dot{x}_j^k, y_j^k, \dot{y}_j^k]^T \in \mathbb{R}^4$ where $\mathbf{p}_j^k = [x_j^k, y_j^k]^T$ is the position vector, $\mathbf{v}_j^k = [\dot{x}_j^k, \dot{y}_j^k]^T$ is the velocity vector and superscript T means transpose. The augmented state of track j is described at time k by vector $\mathbf{r}_j^k = \left[(\mathbf{x}_j^k)^T, i_j^k \right]^T \in \mathbb{R}^4 \times \mathbb{N}$ where i_j^k is the label of track j . It is imposed that $i_1^k < i_2^k < \dots < i_{t^k}^k$ to ensure unique labelling of the tracks. The augmented multitrack state vector is formed by stacking the individual augmented track state vectors into one vector $\mathbf{R}^k = \left[(\mathbf{r}_1^k)^T, \dots, (\mathbf{r}_{t^k}^k)^T \right]^T = \left[(\mathbf{x}_1^k)^T, i_1^k, \dots, (\mathbf{x}_{t^k}^k)^T, i_{t^k}^k \right]^T$. Since the number t^k of targets is unknown, the variable

$$\left(\mathbf{R}^k, t^k \right) \in \bigcup_{t=0}^{\infty} (\mathbb{R}^4 \times \mathbb{N})^t \times \{t\} : i_1^k < i_2^k < \dots < i_t^k.$$

Following [13], [29], the set $\mathbb{R}^0 = \{\theta\}$ where θ is a symbol for no target. A similar convention is used for \mathbb{N}^0 . This is a Bayesian hierarchical model where t^k determines the model (in our case the target

number) of the multitarget state [14]–[16].

The state at time k is observed through the measurement vector \mathbf{z}^k . In a Bayesian framework, estimation of (\mathbf{R}^k, t^k) based on the sequence of measurements $\mathbf{z}^{1:k} = (\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^k)$ up to time k is achieved by the recursive computation of the posterior PDF $p(\mathbf{R}^k, t^k | \mathbf{z}^{1:k}) = p(t^k | \mathbf{z}^{1:k}) p(\mathbf{R}^k | \mathbf{z}^{1:k}, t^k)$ which is normalised such that

$$\sum_{t^k=0}^{\infty} p(t^k | \mathbf{z}^{1:k}) \sum_{i_1^k=1}^{\infty} \dots \sum_{i_{t^k}^k=i_{t^k-1}^k+1}^{\infty} \int d\mathbf{x}_1^k \dots d\mathbf{x}_{t^k}^k p(\mathbf{R}^k | \mathbf{z}^{1:k}, t^k) = 1 \quad (1)$$

The explanation for the normalisation given in (1) is the following. In a Bayesian hierarchical model, firstly, we sum over all the models $t^k \in \{0\} \cup \mathbb{N}$ and then we integrate the state for that model. As the labels are discrete variables a sum is performed rather than an integral. In addition, the sum over the labels in (1) takes into account that $i_1^k < i_2^k < \dots < i_{t^k}^k$, which ensures unique labelling.

The posterior is calculated recursively in two phases: prediction and update [30]. The prediction phase makes use of the Chapman-Kolmogorov equation:

$$\begin{aligned} & p(\mathbf{R}^k, t^k | \mathbf{z}^{1:k-1}) \\ = & \sum_{t^{k-1}=0}^{\infty} \sum_{i_1^{k-1}=1}^{\infty} \dots \sum_{i_{t^{k-1}}^{k-1}=i_{t^{k-1}-1}^{k-1}+1}^{\infty} \int d\mathbf{x}_1^{k-1} \dots d\mathbf{x}_{t^{k-1}}^{k-1} p(\mathbf{R}^k, t^k | \mathbf{R}^{k-1}, t^{k-1}) p(\mathbf{R}^{k-1}, t^{k-1} | \mathbf{z}^{1:k-1}) \end{aligned} \quad (2)$$

The explanation about why (2) has this form is the same as for (1). The update equation is obtained by Bayes' rule:

$$p(\mathbf{R}^k, t^k | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{R}^k, t^k) p(\mathbf{R}^k, t^k | \mathbf{z}^{1:k-1}) \quad (3)$$

It should be noted that PFs only require the posterior PDF up to a normalising constant [30]. Therefore, the normalisation constant is not included in (3) as it is not necessary in the PF implementation explained in the rest of the paper. When $t^{k-1} = 0$, \mathbf{R}^{k-1} can only take the value θ , thus, it is a discrete state so the integrals in (2) come down to a substitution of the probability mass of that state.

In the following, we introduce an equivalent notation that simplifies the writing of the necessary PDFs involved in (2) and (3). Let us define the variable (\mathbf{X}^k, I^k) where $\mathbf{X}^k = [(\mathbf{x}_1^k)^T, \dots, (\mathbf{x}_{t^k}^k)^T]^T \in \mathbb{R}^{4t^k}$ is the multitrack vector and $I^k = \{i_1^k, i_2^k, \dots, i_{t^k}^k\}$ is the set of labels. Now, it is the size of I^k that governs the dimensionality of \mathbf{X}^k . It is shown in [9, Section 4.2] that there is a one-to-correspondence between (\mathbf{X}^k, I^k) and (\mathbf{R}^k, t^k) . This implies that computing $p(\mathbf{X}^k, I^k | \mathbf{z}^{1:k})$ is equivalent to calculating $p(\mathbf{R}^k, t^k | \mathbf{z}^{1:k})$. Additionally, there is also a bijection with an equivalent RFS with labels [9, Appendix B]. Table I indicates the quantities of interest in each one of the equivalent notations. Because of the equivalence among these notations, the paper could have been written using any of them taking into

Table I – Equivalent notations for multitarget tracking with labels

Notation	Quantity of interest
Hierarchical model with labels [13]	$(\mathbf{R}^k, t^k) = \left(\left[(\mathbf{x}_1^k)^T, i_1^k, \dots, (\mathbf{x}_{t^k}^k)^T, i_{t^k}^k \right]^T, t^k \right)$
JMKPD [28]	$(\mathbf{X}^k, I^k) = \left(\left[(\mathbf{x}_1^k)^T, \dots, (\mathbf{x}_{t^k}^k)^T \right]^T, \{i_1^k, i_2^k, \dots, i_{t^k}^k\} \right)$
RFS with labels [2]	$\left\{ \left[(\mathbf{x}_1^k)^T, i_1^k \right]^T, \dots, \left[(\mathbf{x}_{t^k}^k)^T, i_{t^k}^k \right]^T \right\}$

account that the RFS notation uses set densities and set integrals [2]. Nevertheless, this paper is written using the JMKPD for two reasons: using sets for the labels enables us to write the transition PDFs in a more compact form than using the hierarchical model in [13] and using vectors for the multitarget state enables us to describe the PP method in Section III in a more compact form than using sets.

Our aim is to compute $p(\mathbf{X}^k, I^k | \mathbf{z}^{1:k})$ recursively. To this end, we need to calculate (2) and (3) using a target transition PDF $p(\mathbf{X}^k, I^k | \mathbf{X}^{k-1}, I^{k-1})$ and the PDF of the measurement given the multitarget state $p(\mathbf{z}^k | \mathbf{X}^k, I^k)$, which are analogous to their representation using the variable (\mathbf{R}^k, t^k) and can be also written in the RFS framework. These are explained in the following subsections.

A. Multitrack model

This model is mainly the one in [28]. The model takes into account the births, deaths and motions of the targets. We assume that the a priori evolution of the number of targets in track is modelled by an $M/M/\infty$ birth-death process [31] with parameters, λ , the a priori arrival rate of targets, and $1/\mu$, the average life of a target. The a priori arrival rate of targets λ is taken to be proportional to the area A_S of the surveillance region, $\lambda = \lambda_A A_S$, where λ_A would denote the prior arrival of targets per area unit. According to this model, the probability that b^k new targets appear at time k :

$$p(b^k) = \frac{e^{-\lambda\tau} (\lambda\tau)^{b^k}}{b^k!} \quad (4)$$

where τ is the sampling interval [32]. The probability that a target dies in a sampling interval is:

$$p_d = 1 - e^{-\mu\tau} \quad (5)$$

We define the sets: $S^k = \{s_1^k, s_2^k, \dots, s_{t^k-d^k}^k\}$ the set of labels of the targets that survive from time k to $k+1$ where d^k is the number of targets that die at time k and $N^{k+1} = \{n_1^{k+1}, n_2^{k+1}, \dots, n_{b^{k+1}}^{k+1}\}$ the set of labels of the targets that are born at time $k+1$ so $I^{k+1} = S^k \cup N^{k+1}$. We use $\mathbf{X}^{k+1}(S^k)$ to denote the components of the vector \mathbf{X}^{k+1} that correspond with the set of tracks in S^k . That is, $\mathbf{X}^{k+1}(S^k) = \left[(\mathbf{x}^{k+1}(s_1^k))^T, \dots, (\mathbf{x}^{k+1}(s_{t^k-d^k}^k))^T \right]^T$ where $\mathbf{x}^{k+1}(s_j^k) = \left[x^{k+1}(s_j^k), \dot{x}^{k+1}(s_j^k), y^{k+1}(s_j^k), \dot{y}^{k+1}(s_j^k) \right]^T$

represents the state of a target whose label is s_j^k at time $k + 1$ and $s_1^k < s_2^k < \dots < s_{t^k-d}^k$ to ensure unique ordering.

Therefore:

$$p\left(\mathbf{X}^{k+1}, I^{k+1} \mid \mathbf{X}^k, I^k\right) = p\left(\mathbf{X}^{k+1}\left(S^k\right), S^k, \mathbf{X}^{k+1}\left(N^{k+1}\right), N^{k+1} \mid \mathbf{X}^k, I^k\right) \quad (6)$$

Equation (6) comes from the fact that each component of \mathbf{X}^{k+1} is either in $\mathbf{X}^{k+1}\left(S^k\right)$ or $\mathbf{X}^{k+1}\left(N^{k+1}\right)$ and $I^{k+1} = S^k \cup N^{k+1}$. Using the same procedure as in [28], equation (6) can be written as:

$$p\left(\mathbf{X}^{k+1}, I^{k+1} \mid \mathbf{X}^k, I^k\right) = p\left(N^{k+1}\right) p\left(S^k \mid I^k\right) p\left(\mathbf{X}^{k+1}\left(N^{k+1}\right) \mid N^{k+1}\right) p\left(\mathbf{X}^{k+1}\left(S^k\right) \mid \mathbf{X}^k, S^k\right) \quad (7)$$

Now we give the expressions for all the probabilities in (7) and explain their meaning. The first probability indicates how the new labels are assigned. To this end, let us partition the surveillance area $V \subset \mathbb{R}^2$ into M cells of the same size V_1, V_2, \dots, V_M . Assume that at most one target can appear in a given cell at a given sample time. The target appearing in the j th cell at the k th time is given the label $j + (k - 1)M$. Let $F^k = \{(k - 1)M + 1, \dots, kM\}$ denote the set of all possible labels that can be assigned at time k . Then

$$p\left(N^{k+1}\right) \propto \begin{cases} p\left(b^{k+1}\right) & \text{if } N^{k+1} \subseteq F^{k+1} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where we should recall that $|N^{k+1}| = b^{k+1}$. It should be noted that we also use a grid on M sensors in the measurement model to be described in Section II-B. However, this does not imply that this way of assigning the target labels can only be applied to this kind of measurement model. In general, for any measurement model, we can partition the surveillance area such that only one target can appear in a given region at a given time step.

The second probability in (7) indicates the set of labels of the targets that survive time k :

$$p\left(S^k \mid I^k\right) = \begin{cases} \prod_{j=1}^{t^k} \left[\chi_{S^k}\left(i_j^k\right) (1 - p_d) + \left(1 - \chi_{S^k}\left(i_j^k\right)\right) p_d \right] & \text{if } S^k \subseteq I^k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $\chi_S(B) = 1$ if $B \subseteq S$ and zero otherwise and it has been assumed that each target dies with a probability p_d independently from the rest.

The third probability in (7) indicates the a priori state of the targets that are born at time k . We assume that the states of newborn targets are independent and target n_j^k appears in cell $V_{n_j^k}$ with an initial position uniformly distributed in this cell and an initial velocity distributed following a PDF $p_{v_0}(\cdot)$. Therefore

$$p\left(\mathbf{X}^k\left(N^k\right) \mid N^k\right) = \prod_{j=1}^{b^k} U_{V_{n_j^k}}\left(x^k\left(n_j^k\right), y^k\left(n_j^k\right)\right) \cdot p_{v_0}\left(\dot{x}^k\left(n_j^k\right), \dot{y}^k\left(n_j^k\right)\right) \quad (10)$$

where $U_{V_{n_j^k}}(x, y)$ is a uniform random variable in the cell $V_{n_j^k}$, whose area is $A_{n_j^k} = A_S/M$ (A_S is the area of the surveillance region), evaluated at (x, y) .

The fourth probability in (7) is the dynamic model, a nearly constant velocity model [33]:

$$p\left(\mathbf{X}^{k+1}(S^k) \mid \mathbf{X}^k, S^k\right) = \prod_{j=1}^{t^k-d^k} \mathcal{N}\left(\mathbf{x}^{k+1}(s_j^k); \mathbf{F}\mathbf{x}^k(s_j^k), \mathbf{Q}\right) \quad (11)$$

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} \quad (12)$$

$$\mathbf{Q} = \sigma_u^2 \mathbf{I}_2 \otimes \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix} \quad (13)$$

where it is assumed that the targets move independently, $\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{Q})$ is the Gaussian PDF evaluated at \mathbf{x} with mean $\bar{\mathbf{x}}$ and covariance matrix \mathbf{Q} , \mathbf{I}_m is the $m \times m$ identity matrix, \otimes is the Kronecker product and σ_u^2 is the continuous-time process noise intensity [33].

B. Measurement models

We consider two measurement models. The first model is the pixellised image model commonly used in track-before-detect algorithms, where each pixel (sensor) corresponds to the received power in a particular spatial location [3], [6], [7], [34]. This model is used to compare our algorithm with the existence grid [7], [25] as to the creation and deletion of tracks. The other is a centralised sensor network in which the sensors are deployed in a surveillance area, measure the received power from emitting targets and transmit its value to a fusion centre, which carries out detection and tracking tasks [35]. Both models only differ in how the signal-to-noise ratio SNR at each sensor/pixel is calculated.

For the pixellised sensor model, the signal-to noise ratio produced by a target with coordinates $[x, y]^T$ in a pixel (sensor) is [7]:

$$SNR(x, y, x_s, y_s) = \begin{cases} SNR & |x - x_s| \leq l_x/2 \text{ and } |y - y_s| \leq l_y/2 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where $[x_s, y_s]^T$ are the coordinates of the sensor and $[l_x, l_y]^T$ is the size of the pixels.

In the second model, the signal-to-noise ratio produced by a target depends on the distance r from the target to the sensor following a piecewise linear function. The function is defined with $N + 1$ intervals whose endpoints and values are $\{r_0 = 0, r_1, \dots, r_N, r_{N+1} = \infty\}$ and $SNR_{r,i}$ $i \in \{1, \dots, N + 1\}$,

respectively. Therefore:

$$SNR(r) = \begin{cases} SNR_{r,1} & 0 \leq r < r_1 \\ \dots & \dots \\ SNR_{r,N} & r_{N-1} \leq r < r_N \\ 0 & r \geq r_N \end{cases} \quad (15)$$

which can approximate any isotropic propagation model as accurately as possible.

For both models we assume that there are M sensors located in the surveillance area with the vector $\mathbf{z}^k = [z_1^k, z_2^k, \dots, z_M^k]^T$ containing the measurements from each sensor which are independent in each sensor. We assume Rayleigh-distributed measurements. This distribution corresponds to envelope-detected signals under a Gaussian model. This model has been used in radar systems [26] and radio signal propagation [36]. Then, the PDF of the measurement at sensor j conditioned on the current multitarget state depends on the total signal-to-noise ratio at sensor j , SNR_j , which is the sum of the signal-to-noise ratios produced by each target [37], calculated using (14) or (15) depending on the sensor model. Therefore:

$$p(\mathbf{z}^k | \mathbf{X}^k, I^k) = \prod_{j=1}^M p(z_j^k | SNR_j) \quad (16)$$

where the dependency of SNR_j on \mathbf{X}^k is not given explicitly. The PDF of the measurement for sensor j for non-thresholded data is:

$$p(z_j^k | SNR_j) = \frac{z_j^k}{1 + SNR_j} \exp\left(-\frac{(z_j^k)^2}{2(1 + SNR_j)}\right) \quad (17)$$

In the numerical simulations in Section V, we also analyse the effect of thresholding the data on performance as in [6], [7]. This binary data (detections) is obtained using threshold Γ :

$$\Gamma = \sqrt{-2 \ln P_{fa}} \quad (18)$$

where P_{fa} is the false alarm probability per sensor. Then, the PDF of the measurement for sensor j for binary data is:

$$p(z_j^k | SNR_j) = P_{fa}^{1/(1+SNR_j)} \delta(z_j^k - 1) + (1 - P_{fa}^{1/(1+SNR_j)}) \delta(z_j^k) \quad (19)$$

where $\delta(\cdot)$ denotes the Dirac delta. It should be noted that if we use any other measurement model instead of (16), it does not affect the development of the main novelties of the paper: the PP method of Section III and the two-layer PF of Section IV. These measurement models have been used to analyse the performance of the algorithms in Section V.

III. PARTICLE FILTER IMPLEMENTATION FOR A FIXED AND KNOWN NUMBER OF TRACKS

In this section, we present a PF to approximately solve (2) and (3) when the number of tracks is fixed and known. We will generalise the algorithm for a variable and unknown number of tracks in the next section. In this case, the number of tracks over time is constant $t^k = t$, $I^k = I = \{1, \dots, t\}$ the set of labels of the surviving tracks coincides with the labels at the previous step $S^k = I$ and no tracks are born at any time $N^k = \emptyset$. Therefore, the JMCPD comes down to estimating $p(\mathbf{X}^k | \mathbf{z}^{1:k})$ as the labels are implicit in the ordering of the multitarget state vector \mathbf{X}^k . In addition, as nonlinearity in our tracking model enters in the relationship between the measurements and the positions of the tracks and the relationship between the velocities and the positions is linear and Gaussian, the posterior of the velocity elements can be computed exactly using a Kalman filter conditional on the trajectory of the targets. This technique, known as Rao-Blackwellisation, improves the filter performance for a given sample size [7], [38]. Then, the Rao-Blackwellised PF approximation to the JMCPD at time k is approximated as [9]

$$p(\mathbf{X}^k | \mathbf{z}^{1:k}) \approx \sum_{i=1}^{N_{par}} w_i^k \delta(\mathbf{P}^k - \mathbf{P}_i^k) \prod_{j=1}^t p(\mathbf{v}_j^k | \mathbf{p}_{j,i}^{0:k}) \quad (20)$$

where

$$p(\mathbf{v}_j^k | \mathbf{p}_{j,i}^{0:k}) = \mathcal{N}(\mathbf{v}_j^k; \bar{\mathbf{v}}_{j,i}^k, \Sigma_{j,i}^k) \quad (21)$$

and N_{par} is the number of particles, w_i^k is the normalised weight of particle i at time k , \mathbf{P}^k is the vector that contains the positions of all tracks at time k , \mathbf{P}_i^k is the i th particle at time k , and $\bar{\mathbf{v}}_{j,i}^k$ and $\Sigma_{j,i}^k$ are the posterior mean and covariance matrix of the velocity elements conditional on the trajectory of positions for the i th particle and j th track $\mathbf{p}_{j,i}^{0:k}$. At time k , what we need to store is the Rao-Blackwellised particles $\mathbf{X}_i^k = (\mathbf{P}_i^k, \bar{\mathbf{v}}_{1,i}^k, \Sigma_{1,i}^k, \dots, \bar{\mathbf{v}}_{t,i}^k, \Sigma_{t,i}^k)$ for $i = 1, \dots, N_{par}$.

The PF approximation of the posterior density of the positions at time $k+1$ using Bayes' rule and (11) in (20) and integrating out the targets' velocities at time $k+1$ is [9]:

$$p(\mathbf{P}^{k+1} | \mathbf{z}^{1:k+1}) \propto p(\mathbf{z}^{k+1} | \mathbf{P}^{k+1}) \sum_{i=1}^{N_{par}} w_i^k \prod_{j=1}^t p(\mathbf{p}_j^{k+1} | \mathbf{x}_{j,i}^k) \quad (22)$$

where $\mathbf{x}_{j,i}^k$ is the Rao-Blackwellised i th particle of track j at time k and $p(\mathbf{p}_j^{k+1} | \mathbf{x}_{j,i}^k)$ is Gaussian with mean $\mathbf{p}_{j,i}^k + \tau \bar{\mathbf{v}}_{j,i}^k$ and covariance matrix $\tau^2 \Sigma_{j,i}^k + \mathbf{Q}_p$ where \mathbf{Q}_p is the process noise covariance matrix for the position elements according to (13).

Then, the PF approximation of the posterior needs to obtain samples that are distributed according to (22). There are a lot of ways to approximate (22) using importance sampling, each one defining a PF.

We propose an importance density constructed using a method called parallel partition (PP). The method developed here is a convenient, fast way to propagate particles in multitarget situations.

A. Parallel Partition method

Here, we propose the PP method as a modification of the well-known IP method and provide a detailed mathematical explanation on how both methods work via auxiliary sampling. IP and PP methods make the following assumption:

Assumption 1. *Tracks are a priori independent with evenly distributed weights, $w_i^k = 1/N_{par}$, so that*

$$p\left(\mathbf{P}^{k+1} \mid \mathbf{z}^{1:k}\right) \propto \prod_{j=1}^t \sum_{i=1}^{N_{par}} p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,i}^k\right) \quad (23)$$

The proof of (23) under Assumption 1 can be found in the Appendix. Under Assumption 1, the posterior PDF becomes:

$$p\left(\mathbf{P}^{k+1} \mid \mathbf{z}^{1:k+1}\right) \propto p\left(\mathbf{z}^{k+1} \mid \mathbf{P}^{k+1}\right) \prod_{j=1}^t \sum_{i=1}^{N_{par}} p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,i}^k\right) \quad (24)$$

We should note that getting weights evenly distributed at time k can be done by performing a resampling step of the particles at time k [30].

Then, what we propose is to sample an auxiliary variable for each track such that the posterior is:

$$p\left(\mathbf{P}^{k+1}, \mathbf{a} \mid \mathbf{z}^{1:k+1}\right) \propto p\left(\mathbf{z}^{k+1} \mid \mathbf{P}^{k+1}\right) \prod_{j=1}^t p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,a_j}^k\right) \quad (25)$$

where $\mathbf{a} = [a_1, \dots, a_t]^T \in \{1, \dots, N_{par}\}^t$ and a_j indicates that we are choosing the component a_j for track j on the mixture given by the prior, see equation (24). The idea of sampling in a higher dimension has traditionally been used, for instance, in the auxiliary PF, and can provide a lot of benefits [22], [30], [39]. We should note that integrating out the auxiliary variables in (25), we get the PDF of (24) so this definition of auxiliary variables is sound. If we draw a sample from the joint density (25) and then discard the part of the sample that corresponds with \mathbf{a} , then we are producing a sample from (24) as required.

If we draw \mathbf{P}_i^{k+1} , $\mathbf{a}_i = [a_1^i, \dots, a_t^i]^T$ from an importance density $q_1(\cdot)$ for $i = 1, \dots, N_{par}$, the updated weights are calculated as:

$$w_i^{k+1} \propto \frac{p\left(\mathbf{z}^{k+1} \mid \mathbf{P}_i^{k+1}\right) \prod_{j=1}^t p\left(\mathbf{p}_{j,i}^{k+1} \mid \mathbf{x}_{j,a_j^i}^k\right)}{q_1\left(\mathbf{P}_i^{k+1}, \mathbf{a}_i \mid \mathbf{z}^{1:k+1}\right)} \quad (26)$$

The importance density in the PP filter is taken to be:

$$q_1\left(\mathbf{P}^{k+1}, \mathbf{a} \mid \mathbf{z}^{1:k+1}\right) = \prod_{j=1}^t b_j\left(\mathbf{p}_j^{k+1}\right) p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,a_j}^k\right) \quad (27)$$

where

$$b_j \left(\mathbf{p}_j^{k+1} \right) \propto p \left(\mathbf{z}^{k+1} \mid \hat{\mathbf{p}}_1^{k+1|k}, \dots, \hat{\mathbf{p}}_{j-1}^{k+1|k}, \mathbf{p}_j^{k+1}, \hat{\mathbf{p}}_{j+1}^{k+1|k}, \dots, \hat{\mathbf{p}}_t^{k+1|k} \right) \quad (28)$$

and $\hat{\mathbf{p}}_j^{k+1|k}$ is the predicted position of track j averaged over all the particles. We call $b_j \left(\mathbf{p}_j^{k+1} \right)$ the first-stage weights. In contrast, we want to indicate that the IP method uses $b_j \left(\mathbf{p}_j^{k+1} \right) \propto p \left(\mathbf{z}^{k+1} \mid \mathbf{p}_j^{k+1} \right)$ rather than (28).

Substituting (27) and (28) into (26), the unnormalised particle weights, second-stage weights, are:

$$w_i^{k+1} \propto \frac{p \left(\mathbf{z}^{k+1} \mid \mathbf{p}_i^{k+1} \right)}{\prod_{j=1}^t b_j \left(\mathbf{p}_{j,i}^{k+1} \right)} \quad (29)$$

The computational expense of evaluating (28) and then (29), can be lowered if we separate the tracks into $s \leq t$ clusters $C_1^{k+1}, \dots, C_s^{k+1}$ such that $\bigcup_{l=1}^s C_l^{k+1} = I$ and $\forall l \in \{1, \dots, s\}, \forall i \in C_l^{k+1}$

$$\left\| \hat{\mathbf{p}}_i^{k+1|k} - \hat{\mathbf{p}}_j^{k+1|k} \right\| \leq \Gamma_c \rightarrow j \in C_l^{k+1} \quad (30)$$

where Γ_c is a threshold. Equation (30) is a mathematical way to indicate that if the distance between the predicted positions of two tracks is below Γ_c , then, these tracks belong to the same cluster¹. Using the same reasoning as in [7], for a sufficiently large value of Γ_c , we can write (28) as:

$$b_j \left(\mathbf{p}_j^{k+1} \right) \propto p \left(\mathbf{z}^{k+1} \mid \hat{\mathbf{C}}_{l-\{j\}}^{k+1|k}, \mathbf{p}_j^{k+1} \right) \quad (31)$$

where $\hat{\mathbf{C}}_{l-\{j\}}^{k+1|k}$ are the predicted positions of the tracks that belong to cluster l , in which track j is also included, except the position of track j averaged over all the particles.

In the rest of the section, we explain how we draw samples from (27) and provide a discussion about the proposed method.

1) *Sampling from the importance density:* Here, we explain how IP and PP draw samples from (27) recalling that the only difference between both methods is the definition of $b_j \left(\mathbf{p}_j^{k+1} \right)$. We can write

$$q_1 \left(\mathbf{P}^{k+1}, \mathbf{a} \mid \mathbf{z}^{1:k+1} \right) = \prod_{j=1}^t q_{1,j} \left(\mathbf{p}_j^{k+1}, a_j \mid \mathbf{z}^{1:k+1} \right) = \prod_{j=1}^t b_j \left(\mathbf{p}_j^{k+1} \right) p \left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,a_j}^k \right) \quad (32)$$

where $q_{1,j}(\cdot)$ is the importance density to draw positions and indices for track j . It should be noted that it is generally not possible to sample directly from $q_{1,j}(\cdot)$. Instead, we use the approximate method sampling/importance resampling in [40] to get samples from $q_{1,j}(\cdot)$.

¹A track can belong to only one cluster and the computational burden of clustering is almost negligible in the algorithm.

a) *General algorithm:* According to [40], we can define a “first-pass approximation” of $q_{1,j}(\mathbf{p}_j^{k+1}, a_j | \mathbf{z}^{1:k+1})$ called $h(\mathbf{p}_j^{k+1}, a_j | \mathbf{z}^{1:k+1})$. Then, we draw M_{par} samples from $h(\mathbf{p}_j^{k+1}, a_j | \mathbf{z}^{1:k+1})$, which are denoted by $\mathbf{p}_{j,p}^*, a_{j,p}^*$ for $p = 1, \dots, M_{par}$, and we evaluate

$$r(\mathbf{p}_{j,p}^*, a_{j,p}^*) \propto \frac{q_{1,j}(\mathbf{p}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k+1})}{h(\mathbf{p}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k+1})} \quad (33)$$

Finally, we draw N_{par} values of $(\mathbf{p}_j^{k+1}, a_j)$ from the distribution defined by $(\mathbf{p}_{j,p}^*, a_{j,p}^*)$ for $p = 1, \dots, M_{par}$ with probability proportional to $r(\mathbf{p}_{j,p}^*, a_{j,p}^*)$. The resulting particles are samples from $q_{1,j}(\cdot)$ [40].

b) *Application to IP and PP:* In both IP and PP methods, $h(\cdot)$ is

$$h(\mathbf{p}_j^{k+1}, a_j | \mathbf{z}^{1:k+1}) \propto p(\mathbf{p}_j^{k+1} | \mathbf{x}_{j,a_j}^k) \quad (34)$$

and IP and PP sample from $h(\cdot)$ via stratified sampling [41]. The sample space of $(\mathbf{p}_j^{k+1}, a_j)$ is $\Omega = \mathbb{R}^2 \times \{1, \dots, N_{par}\}$. Then, we can define N_{par} disjoint subregions $\{D_1, \dots, D_{N_{par}}\}$ such that $D_i = \mathbb{R}^2 \times \{i\}$ and $\Omega = \bigcup_{i=1}^{N_{par}} D_i$. From (34), the probability that a sample belongs to region D_i is $1/N_{par}$, i.e., the regions are equally probable. Then, assuming we want M_{par} samples from $h(\cdot)$, there are M_i samples in region D_i such that $\sum_{i=1}^{N_{par}} M_i = M_{par}$. Since the variance in these regions is the same, the optimal way to assign M_i is $M_i = M_{par}/N_{par}$ [41]. In the IP and PP method $M_{par} = N_{par}$, so $M_i = N_{par}/N_{par} = 1$ although it can be easily generalised for $M_{par} > N_{par}$. This implies that sampling from $h(\mathbf{p}_j^{k+1}, a_j | \mathbf{z}^{1:k+1})$ can be done going through all the regions (i.e. indices of the particles), drawing one position per each region (particle), noting that in each region the index is deterministic.

Once we have M_{par} samples from (34), we draw N_{par} values of $(\mathbf{p}_j^{k+1}, a_j)$ from the empirical distribution defined over the values $(\mathbf{p}_{j,p}^*, a_{j,p}^*)$ for $p = 1, \dots, N_{par}$ with probabilities proportional to $r(\mathbf{p}_{j,p}^*, a_{j,p}^*) = b_j(\mathbf{p}_{j,p}^*)$ getting samples from $q_{1,j}(\cdot)$. These are the steps shown in Table II. We should mention that in [18] and [6] they also sample following the steps in Table II but they do not justify why.

2) *Discussion about the PP method:* The important features of the PP method are subparticle crossover and the way the first-stage weights are chosen. By “subparticle” we refer to the part of a particle that represents a single track. Therefore, a particle is made up of the subparticles of all the tracks. Then, by sampling an auxiliary variable for each track, \mathbf{a} in (27), what we are doing is that a particle at time step k is formed by the propagation of subparticles that belonged to different particles at time $k-1$, see Figure 1. For example, a certain $\mathbf{a}_i = [a_1^i, \dots, a_t^i]^T$ indicates that to form particle i we are using subparticle a_1^i for target one, a_2^i for target two and so on, therefore, mixing them and doing subparticle crossover.

Table II – Independent and Parallel Partition Subroutines

IP and PP subroutines for track or pre-track j with Rao-Blackwellisation

- For each particle $i = 1, \dots, N_{par}$ (Region D_i):
 - Get a sample for the position $\mathbf{p}_{j,i}^* \sim p(\mathbf{p}_j^{k+1} | \mathbf{x}_{j,i}^k)$ using (11).
 - Calculate the mean and covariance matrix of the velocity using a Kalman filter $\bar{\mathbf{v}}_{j,i}^*$ y $\Sigma_{j,i}^*$.
 - IF we are using IP
 - * Calculate $b_j(\mathbf{p}_{j,i}^*) = p(\mathbf{z}^{k+1} | \mathbf{p}_{j,i}^*)$ using (16).
 - ELSE (we are using PP)
 - * Calculate $b_j(\mathbf{p}_{j,i}^*) = p(\mathbf{z}^{k+1} | \mathbf{p}_{j,i}^*, \hat{\mathbf{C}}_{l-\{j\}}^{k+1|k})$ using (16) where l is the cluster of track j .
- Normalise $b_j(\mathbf{p}_{j,i}^*)$ to sum to one over $i = 1, \dots, N_{par}$.
- Resampling step for each track. For each particle $i = 1, \dots, N_{par}$:
 - Sample an index p from the distribution defined by $b_j(\mathbf{p}_{j,i}^*)$ over $l = 1, \dots, N_{par}$.
 - Set $\mathbf{x}_{j,i}^{k+1} = \{\mathbf{p}_{j,p}^*, \bar{\mathbf{v}}_{j,p}^*, \Sigma_{j,p}^*\}$ and the parent's index $a_j^i = p$.

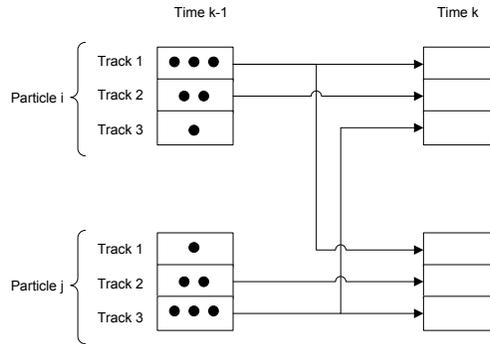


Figure 1 – Subparticle crossover for three tracks: A particle at time k can be made up of the propagation of the tracks that belonged to different particles at time $k - 1$. The circles inside each subparticle stand for the first-stage weight of the subparticle, see (31). At time k , we tend to form particles with subparticles that have a high first-stage weight. In this case, $\mathbf{a}_i = [i, i, j]^T$ and $\mathbf{a}_j = [i, j, j]^T$.

Subparticle crossover is also included in the IP method [6], [8], [19]. In fact, as we indicated, the IP method is equivalent to the PP method if the first stage weights are given by $b_j(\mathbf{p}_j^{k+1}) \propto p(\mathbf{z}^{k+1} | \mathbf{p}_j^{k+1})$ rather than (31). However, IP method performance plummets when there are several targets in close proximity. This loss of performance is due to the way the first-stage weights are chosen as only one target is considered. This leads to the selection of subparticles that are appropriate for representing one target because the likelihood for that subparticle considering one target is high. Nonetheless, this does not imply that a particle formed by stacking good subparticles for representing one target will have a

high likelihood when considering all the targets together.

It should be noted that Assumption 1, which is required for subparticle crossover, does not necessarily hold for a group of closely spaced tracks as their positions conditioned on the measurements could be dependent. However, assumptions of this kind are quite common in multitarget tracking algorithms. For instance the well-known joint probabilistic data association filter (JPDAF) [42] assumes a priori independence of the target states as does the Monte Carlo approximation to the JPDAF proposed in [23]. These methods have been shown to work very well despite the occasional invalidity of this assumption. Using Assumption 1 when it is not satisfied can be interpreted as a way of trading-off bias and variance. In particular, we argue that the bias incurred in assuming a priori independence when it does not actually exist is less than the variance which results by the joint sampling of several track states, particularly when tracks states are sampled from the prior and the number of particles is relatively small. This was proved in [43] for the case of a Gaussian posterior.

We also want to highlight that the PP method can be used for any measurement model as it only requires Assumption 1, which is independent of the measurement model. Here, PP has been explained using Rao-Blackwellisation but this is not an essential part of the algorithm. As in Bayesian filtering the likelihood $p(\mathbf{z}^{k+1} | \mathbf{X}^{k+1})$ is always available, PP can always be used as it only requires to sample from the prior and evaluate the likelihood as indicated previously.

IV. PARTICLE FILTER IMPLEMENTATION FOR A VARIABLE AND UNKNOWN NUMBER OF TRACKS

In the previous section we presented the PP method as a way to propagate particles in a multitarget tracking set-up when the number of targets is fixed and known. In this section, we present a two-layer PF to deal with the problem of tracking a variable and unknown number of targets. One layer deals with the termination of the tracks and the tracking itself while the other deals with track initiation. Both layers are PFs on their own, therefore, they are called track PF and pre-track PF, respectively. This idea is illustrated in Figure 2.

Firstly, we describe the importance density for the track PF. The importance density is chosen so that it factorises as:

$$q\left(\mathbf{P}^{k+1}, I^{k+1}, \mathbf{a}, \mathbf{c} \mid \mathbf{z}^{1:k+1}\right) = q_p\left(\mathbf{P}^{k+1}\left(S^k\right), S^k, \mathbf{a} \mid \mathbf{z}^{1:k+1}\right) q_3\left(\mathbf{P}^{k+1}\left(N^{k+1}\right), N^{k+1}, \mathbf{c} \mid \mathbf{z}^{1:k+1}\right) \quad (35)$$

where $\mathbf{a} = [a_1, \dots, a_{t^k}]^T$ is the auxiliary vector of the PP method and, $\mathbf{c} = [c_1, \dots, c_{b^{k+1}}]^T \in \{0, \dots, N_{par}\}^{b^{k+1}}$ is an auxiliary vector used to create new tracks. The use of the variable \mathbf{c} is explained in Section IV-B. The

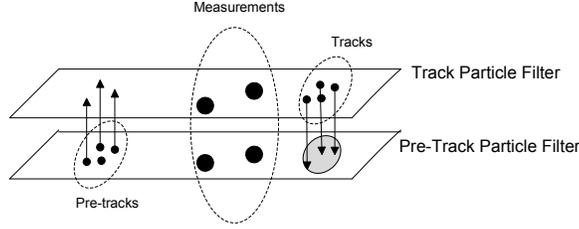


Figure 2 – Two-layer PF diagram. The measurements update the two layers simultaneously. When the pre-tracks are considered tracks, they pass to the upper layer. In the area where there is a track, we do not initiate pre-tracks.

importance density $q_p(\cdot)$ propagates the labels of the surviving targets and their states. The importance density $q_3(\cdot)$ proposes the positions of the tracks that are born at time $k + 1$ and assigns their labels.

The importance density $q_p(\cdot)$ is taken to be:

$$q_p\left(\mathbf{P}^{k+1}\left(S^k\right), S^k, \mathbf{a}\left|\mathbf{z}^{1:k+1}\right.\right) = q_1\left(\mathbf{P}^{k+1}, \mathbf{a}\left|\mathbf{z}^{1:k+1}\right.\right) q_2\left(S^k\left|\mathbf{P}^{k+1}, \mathbf{a}, \mathbf{z}^{1:k+1}\right.\right) \quad (36)$$

where $q_1(\cdot)$ is the importance function of the PP method for known number of targets, equation (27), that only depends on the particles at time k , and $q_2(\cdot)$ determines the survival targets at time k and is defined as:

$$q_2\left(S^k\left|\mathbf{P}^{k+1}, \mathbf{a}, \mathbf{z}^{1:k+1}\right.\right) = \prod_{j=1}^{t^k} \left[q_{2,j}\left(i_{j,a_j}^k \in S^k\left|\mathbf{p}_{1,a_1}^{k-N_d+2:k+1}, \dots, \mathbf{p}_{t^k,a_{t^k}}^{k-N_d+2:k+1}, \mathbf{z}^{k-N_d+2:k+1}\right.\right) \chi_{S^k}\left(i_{j,a_j}^k\right) + \left(1 - q_{2,j}\left(i_{j,a_j}^k \in S^k\left|\mathbf{p}_{1,a_1}^{k-N_d+2:k+1}, \dots, \mathbf{p}_{t^k,a_{t^k}}^{k-N_d+2:k+1}, \mathbf{z}^{k-N_d+2:k+1}\right.\right)\right) \left(1 - \chi_{S^k}\left(i_{j,a_j}^k\right)\right) \right] \quad (37)$$

where $q_{2,j}\left(i_{j,a_j}^k \in S^k\right)$ represents the probability that track i_{j,a_j}^k survives, $\mathbf{p}_{j,a_j}^{k-N_d+2:k+1}$ represents the trajectory of particle a_j of target j from time $k - N_d + 2$ to $k + 1$. Thus, N_d is the number of past time steps that we use to decide if a track survives. We make $q_2(\cdot)$ dependent on several time instants so that we take into account the target's dynamics to remove targets. In Section IV-A, we explain this importance function thoroughly.

Finally, the importance density for sampling the states of newborn tracks $q_3\left(\mathbf{P}^{k+1}\left(N^{k+1}\right), N^{k+1}, \mathbf{c}\left|\mathbf{z}^{1:k+1}\right.\right)$ does not depend on previous targets' positions and we use another PF to draw samples from it, i.e., it is the second layer of the overall PF. In Section IV-B, we address this importance density.

A. Importance density for removing tracks

The removal of a track in a particle is performed via a likelihood ratio test (LRT) [44] based on N_d time instants as indicated by (37). The decision to remove a track considers several time instants so that we take into account target dynamics. The two hypotheses for the test are H_1 : the track i_j^k is alive and H_0 : the track i_j^k is dead. In [28], the algorithm considered each target alone in the surveillance area but that is a simplification because the decision to remove a track should take into account nearby targets. Due to the fact that the importance density has to be non-zero wherever the prior is, we need to define an acceptance probability for the result of the test:

$$q_{2,j} \left(i_j^k \in S^k \mid \mathbf{P}^{k-N_d+2:k+1}, \mathbf{z}^{k-N_d+2:k+1} \right) = \begin{cases} P_a & \Lambda_d = H_1 \\ 1 - P_a & \Lambda_d = H_0 \end{cases} \quad (38)$$

where P_a is the probability of accepting the hypothesis given by the test² and Λ_d is the LRT [44], assuming that track i_j^k belongs to cluster l^k at time k :

$$\frac{\prod_{l=-1}^{N_d-2} p \left(\mathbf{z}^{k-l} \mid \mathbf{C}_{l^{k-l}}^{k-l} \right)}{\prod_{l=-1}^{N_d-2} p \left(\mathbf{z}^{k-l} \mid \mathbf{C}_{l^{k-l} - \{i_j^k\}}^{k-l} \right)} \begin{matrix} \Lambda_d = H_1 \\ \geq \eta_d \\ \Lambda_d = H_0 \end{matrix} \quad (39)$$

where η_d is a threshold, $\mathbf{C}_{l^k}^k$ represents the positions of the tracks in cluster l^k at time k and $\mathbf{C}_{l^k - \{i_j^k\}}^k$ represents the positions of the tracks in cluster l^k at time k except the position of track i_j^k .

B. Importance density for sampling newborn track states. Pre-track particle filter

As we stated previously, the importance density for sampling the states of the newborn tracks, $q_3(\cdot)$ in (35), is built by means of a subsidiary PF. This PF is referred to as pre-track PF because it deals with the initialisation of tracks. The pre-track PF approximates $p \left(\tilde{\mathbf{X}}^k, \tilde{I}^k \mid \mathbf{z}^{1:k} \right)$ where $\tilde{\mathbf{X}}^k$ and \tilde{I}^k stand for the multitrack state vector and set of labels of the pre-tracks. Based on an LRT, it is decided which of these pre-tracks are real and, consequently, are used to build the importance density $q_3(\cdot)$. As it is another PF, we must specify the a priori model, the measurement model and the importance density of the pre-tracks as done with the tracks in Section II. The measurement model and the state evolution model are the same as in Section II except for the births and deaths of pre-tracks, equations (4) and (9). This is explained in the following.

²We mainly use P_a for theoretical reasons but we actually accept the output of the LRT in the simulations. This could be interpreted as $P_a \approx 1$.

1) *Birth and death of pre-tracks*: The initialisation of pre-tracks in a sensor region is done as follows. First, that sensor and its neighbours must not have tracks or pre-tracks in their area of influence (area in which a target produces an SNR above zero). Second, if we are using binary sensors, there has to be a detection in that sensor or if we are using non-thresholded data sensors, the measurement in that sensor has to be higher than the threshold in (18) given P_{fa} . In this subsection, we name these two events as false alarms for the sake of clarity. Hence, for non-thresholded and binary data, P_{fa} is the probability of initiating a pre-track in a sensor which does not have any targets in its surroundings. It should be noted that if we do not use a grid-type measurement model as the ones used in this paper, we could devise a similar procedure to initiate pre-tracks using the regions for labelling described in Section II.

A pre-track can be initiated because of two independent events: the arrival of a real target, modelled by (4) and the occurrence of a false alarm. The probability mass function of the number of false alarms $n_{P_{fa}}$ in each sampling period is a binomial distribution that, under the assumption that $P_{fa} \ll 1$ and $M \gg 1$, where M is the number of sensors, can be approximated by a Poisson random variable [28]:

$$p(n_{P_{fa}}) \approx \frac{e^{-M \cdot P_{fa}} (M \cdot P_{fa})^{n_{P_{fa}}}}{n_{P_{fa}}!} \quad (40)$$

Then, the number of arrivals of pre-tracks in a time interval τ is the sum of two independent Poisson random variables, given by (4) and (40). The result is another Poisson random variable whose parameter is the sum of the parameters. Therefore, the probability that there are \tilde{b}^k births of pre-tracks is

$$p(\tilde{b}^k) \approx \frac{e^{-\tilde{\lambda}} (\tilde{\lambda})^{\tilde{b}^k}}{\tilde{b}^k!} \quad (41)$$

where $\tilde{\lambda} = \lambda \cdot \tau + M \cdot P_{fa}$.

It is considered for simplicity that the pre-tracks have a deterministic life equal to N_b instants of time.

2) *Importance density for the pre-track PF*: As with the track PF, the importance density of the pre-track PF is split into three independent parts: dynamics, births and deaths of pre-tracks. We make the following assumption:

Assumption 2. *Newly appearing targets are well separated.*

Due to Assumption 2, we use the IP method as the importance density for the evolution of the model $\tilde{q}_1(\cdot)$. In view of the deterministic life of a pre-track, there is no need to define an importance density $\tilde{q}_2(\cdot)$ because the pre-tracks are simply removed from the pre-track filter when they have lived for N_b time instants. The importance density $\tilde{q}_3(\cdot)$ used for the births of new pre-tracks works as follows: every time there is a detection in a cell and this cell and its neighbouring cells are not occupied by a track

or a pre-track, a new pre-track is initiated with a position drawn from a uniform PDF in the area of the cell and the target velocity is initiated in a similar way as in Section V.C of [7]: When a pre-track is added to a particle of the pre-track layer, we draw two integers η_1, η_2 such that $P(\eta_i = j) = 1/2$ for $j \in \{-1, 1\}$ $i \in \{1, 2\}$ and then select the initial distribution of the velocity to be Gaussian with mean $[\eta_1, \eta_2]^T$ and covariance matrix $9\mathbf{I}_2$.

3) *Importance density for newborn tracks*: Once a pre-track has lived for N_b time instants, an LRT is carried out for each subparticle of that pre-track to decide whether it is a real track or not. The two hypotheses are H_1 : the pre-track is a real track and H_0 : the pre-track is a false track. Under Assumption 2, this test becomes:

$$\frac{\prod_{l=-1}^{N_b-2} p(\mathbf{z}^{k-l} | H_1, \tilde{\mathbf{x}}^{k-l}(\tilde{i}_j^k))}{\prod_{l=-1}^{N_b-2} p(\mathbf{z}^{k-l} | H_0)} \underset{\Lambda_b = H_0}{\overset{\Lambda_b = H_1}{\gtrless}} \eta_b \quad (42)$$

where η_b is the threshold and $\tilde{\mathbf{x}}^{k-l}(\tilde{i}_j^k)$ denotes the state of the pre-track of label \tilde{i}_j^k at time $k-l$.

Recall from (35) that sampling the states of the newborn tracks requires an auxiliary variable $\mathbf{c} = [c_1, \dots, c_{b^{k+1}}]^T \in \{0, \dots, N_{par}\}^{b^{k+1}}$ where b^{k+1} is the number of pre-tracks in which at least one subparticle has passed the test (42) at time $k+1$. The variable $c_j = 0$ if the j th pre-track is discarded and $c_j \neq 0$ indicates which subparticle from the pre-track PF is used to sample the state of a newborn track in the track PF.

The importance density $q_3(\cdot)$ is taken to be:

$$q_3(\mathbf{P}^{k+1}(N^{k+1}), N^{k+1}, \mathbf{c} | \mathbf{z}^{1:k+1}) = \prod_{j=1}^{b^{k+1}} q_{3,1}(c_j | \mathbf{z}^{1:k+1}) q_{3,2}(\mathbf{p}_j^{k+1} | \mathbf{z}^{1:k+1}, c_j) q_{3,3}(n_j^{k+1} | \mathbf{p}_j^{k+1}) \quad (43)$$

where

$$q_{3,1}(c_j = i | \mathbf{z}^{1:k+1}) = \begin{cases} d_{j,i}/N_{par} & i \neq 0 \\ 1 - \sum_{i=1}^{N_{par}} d_{j,i}/N_{par} & i = 0 \end{cases} \quad (44)$$

$$q_{3,2}(\mathbf{p}_j^{k+1} | \mathbf{z}^{1:k+1}, c_j) \propto p(\mathbf{z}^{k+1} | \mathbf{p}_j^{k+1}) p(\mathbf{p}_j^{k+1} | \tilde{\mathbf{x}}_{j,c_j}^k) \quad c_j > 0 \quad (45)$$

where $\tilde{\mathbf{x}}_{j,i}^k$ is the i th Rao-Blackwellised particle for pre-track j and $d_{j,i}$ is the output of LRT (42) for pre-track j and particle i , i.e., $d_{j,i} = 1$ if $\Lambda_b = H_1$ or $d_{j,i} = 0$ otherwise. When $c_j = 0$, it means that no target is born and, thus, $q_{3,2}(\cdot)$ assigns $\mathbf{p}_j^{k+1} = \theta$, where θ is the symbol for no target [13]. Once we have sampled c_j for all the particles, the position of the new born target is sampled from $q_{3,2}(\cdot)$ using the approximation in [40], i.e., via the IP method in Table II. In addition, $q_{3,3}(\cdot)$ assigns the labels of these newborn tracks. We should note that according to the prior, see (8), the label of a target is deterministic

and depends on the position of the new born target and the time index. Therefore, $q_{3,3}(\cdot)$ corresponds with the prior assuming that the target positions in all the particles lie in the same cell so that the target has the same label in all the particles.

One should note that the probability of transferring pre-track j to a particle in the track PF is $q_{3,1}(c_j > 0 | \mathbf{z}^{1:k+1}) = \sum_{i=1}^{N_{par}} d_{j,i} / N_{par} = n_{pass} / N_{par}$ where n_{pass} is the number of particles that have passed the test (42). Therefore, the more particles there are that pass the test, the more likely it is to initiate a track. In addition, the mean and covariance matrix of the initial velocity of a track are given by a Kalman filter that uses the past positions of the corresponding pre-track.

Then, using the prior model of evolution of targets and the measurement model in Section II and employing (35), (36), (38) and (43), the updated particle weights of the track PF are:

$$w_i^{k+1} \propto \frac{p(\mathbf{z}^{k+1} | \mathbf{P}_i^{k+1}) p(S_i^k | I_i^k) p(N_i^{k+1}) p(\mathbf{P}_i^{k+1} (N_i^{k+1}) | N_i^{k+1})}{q_2(S_i^k) q_3(\mathbf{P}_i^{k+1} (N_i^{k+1}), N_i^{k+1}, \mathbf{c}_i) \prod_{j=1}^{t_i^k} b_j(\mathbf{p}_{j,i}^{k+1})} \quad (46)$$

where the conditioning of the importance densities $q_2(\cdot)$ and $q_3(\cdot)$ has been removed to simplify the notation and S_i^k and N_i^{k+1} are the sets that contain the labels of the surviving targets and the new born targets of particle i , respectively. One should note that the prior PDF of initial target positions, given by (10), is inversely proportional to $A_S^{b^{k+1}}$. On the other hand, $p(N_i^{k+1}) \propto A_S^{b^{k+1}}$, recalling from Section II that $\lambda = \lambda_A A_S$. Then, both factors cancel each other out and the update of the weights is independent of the area of the surveillance region. Finally, a summary of the integrated detection and tracking algorithm is given in Table III.

V. SIMULATION RESULTS

In this section, we compare the proposed algorithm with other methods available in the literature. Firstly, we compare the PP method to other importance densities in the tracking part of the filter. Secondly, we compare the two-layer PF to other methods as for the estimation of the number of targets.

A. Tracking performance

The PP method is compared with other methods available in the literature: the above-mentioned IP, the adaptive proposal method that uses IP when the targets are far from each other and the coupled partition method (CP) when they are in close proximity³ [6], [8], a jointly auxiliary PF (JA) [39], [45], an adaptive auxiliary PF (AA) and the unscented Kalman filter (UKF) [46]. By JA, we denote the traditional auxiliary

³In this paper, we denote by CP the adaptive proposal that makes use of the clusters in [6].

Table III – Proposed algorithm for detection and tracking of multiple targets

- Separate the tracks into clusters using (30).
- For each track, sample the multiple track state using the PP method assuming all the targets survive, see Table II.
- Calculate the quotient of the LRT for the tracks using (39).
- For each pre-track, sample the multiple pre-track state using the IP method, see Table II.
- Calculate the quotient of the LRT for the pre-tracks using (42).
- For all the pre-tracks that have lived N_b time instants, evaluate the test (42).
 - If there are pre-tracks that pass the test, sample the states of the newborn tracks using (43).
- For each particle in the track PF:
 - Remove the tracks by evaluating the importance density (38).
 - Update the weights of the particles using (46).
- Add pre-tracks to the pre-track PF according to $\tilde{q}_3(\cdot)$, see Section IV-B.
- Update the weights of the particles in the pre-track PF, see Section IV-B.
- Perform resampling to obtain an evenly weighted particle set for both layers.

PF, i.e., at each time step we predict the multitarget state at the following time step, we compute the first-stage weights, we perform resampling and sample the multitarget state, see [39], [45] for details. JA never performs subparticle crossover. By AA, we denote an auxiliary PF that works by clustering of closely-spaced targets as in PP. Then, for each cluster we draw the particles using an auxiliary PF. Therefore, AA performs subparticle crossover for targets that are far, i.e., they do not belong to same cluster, and does not for targets that are in the same cluster. The properties of the importance densities of these PFs can be found in Table IV. The implementation of CP in our simulations uses $R = 10$ possible realisations of the future state as in [6]. The PFs have been implemented following the steps in Table III. The only part that changes is the importance density for sampling the surviving targets' states. Our implementation of the UKF also makes use of the clusters. When the targets are isolated, we update their states and covariance matrices independently and when they belong to the same cluster we update all of them jointly. Thus, the number of sigma points that are needed to match the first two moments [46] is $8N_c + 1$ where N_c is the number of targets in the cluster⁴. A more detailed explanation of how the UKF can be used for the kind of measurement model we use in this paper is given in [45]. Our UKF implementation applies the LRT to the mean for track creation and removal. In addition, we have also implemented the algorithm described in [3], in which the posterior is approximated using

⁴The dimension of the state of a target is 4. Consequently, the dimension of the state of a cluster is $4N_c$ and therefore the number of sigma points is $2 \cdot 4N_c + 1$.

Table IV – Properties of the importance densities

	Subparticle crossover (far)	Subparticle crossover (near)	Considers nearby targets
IP	✓	✓	×
CP	✓	×	×
PP	✓	✓	✓
AA	✓	×	✓
JA	×	×	✓

Table V – Simulation parameters

A priori model		Filter (binary)		Filter (non-thresholded)	
λ_A	$1/(100 \cdot \tau \cdot A_S)$	N_d	4	N_d	4
$1/\mu$	$(133 \cdot \tau)$	η_d	$5 \cdot 10^{-3}$	η_d	e^{-10}
σ_u	$5 \text{ m/s}^{3/2}$	N_b	3	N_b	3
		η_b	10^7	η_b	e^7

multi-Bernoulli random finite sets (MBRFS). The implementation of MBRFS in our simulations removes targets with existence probabilities below 0.001 and the initial existence probability for a target is 0.005. Note that other methods for multiple target tracking can only deal with thresholded measurements that require a data association step such as the algorithms given in [23], [47]–[51]. These algorithms cannot be directly applied to the measurement models described in Section II-B and, therefore, we do not provide comparisons with them.

The scenario used to evaluate the performance of the algorithms consists of four targets whose trajectories cross at the same time, see Fig. 3. The sampling period of the trajectories is $\tau = 0.5 \text{ s}$ and there are $l = 130$ time steps in the simulation. The targets appear at different times. The parameters used in the simulations are shown in Table V. The surveillance area is a rectangle whose dimensions are $5000 \text{ m} \times 2500 \text{ m}$. In this area, the sensors are arranged forming a grid of 50×25 sensors whose side is 100 m . Each cell of the grid is one of the regions V_1, \dots, V_M defined in Section II-A. The SNR of a sensor produced by one target is approximated by a piecewise linear function:

$$SNR(r) = \begin{cases} 20 \text{ dB} & r < 60 \text{ m} \\ 10 \text{ dB} & 60 \text{ m} \leq r < 80 \text{ m} \\ 0 & r \geq 80 \text{ m} \end{cases} \quad (47)$$

where r is the distance from the target to the sensor. We also use $P_{fa} = 2 \cdot 10^{-5}$.

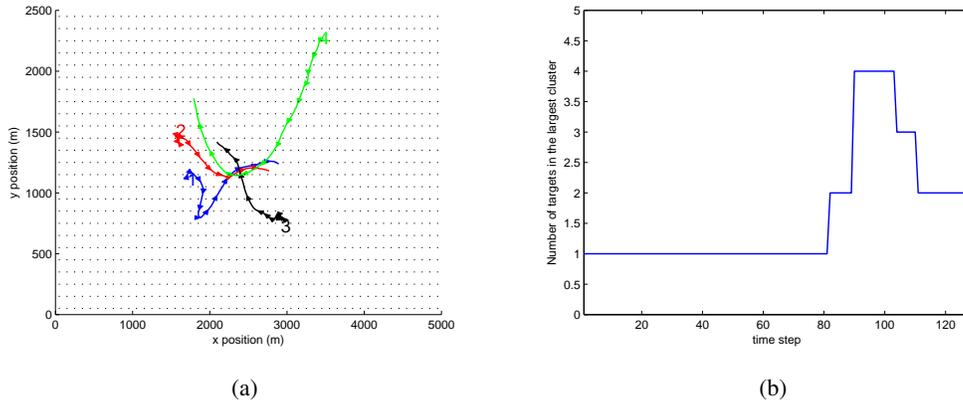


Figure 3 – Simulation scenario: (a) target trajectories and (b) the largest cluster size for the true target positions plotted against time for a clustering threshold $\Gamma_c = 200$ m. In (a), each target is identified at the beginning of its trajectory by a number and there is an arrow to indicate the target position every ten time steps. Target one appears at time step 1, targets two and four at time step 3 and target three at time step 4. The targets do not disappear until the end of the simulation. Black dots indicate the sensor positions.

We evaluate the tracking performance calculating the root mean square (RMS) of the position error by a Monte Carlo simulation with $m = 50$ realisations. To assess the tracking performance, we make a *Track-to-truth Assignment Table* [26] in which each track is associated with one target or with none at each time step depending on the distance from the track to the real position of the target and a threshold that is taken to be $\Lambda = 120$ m. If the distance from a track to a target exceeds that threshold, the track cannot be associated with that target. The threshold for forming the clusters is $\Gamma_c = 200$ m. In addition, a track is assumed to be active if at least 70% of the particles contain it.

Let $\hat{\mathbf{p}}_i^k(j)$, $i = 1, \dots, t^k$, $k = 1, \dots, l$, $j = 1, \dots, m$ denote the position estimate of the i th target on the j th Monte Carlo realisation at time k , $e_i^k(j) = \|\mathbf{p}_i^k - \hat{\mathbf{p}}_i^k(j)\|_2$ denote the position error of the i th target on the j th Monte Carlo realisation at time k and $\chi_{i,j,k} = 1$ if $e_i^k(j) < \Lambda$ and zero otherwise. The variable $\chi_{i,j,k}$ indicates if the i th target is in track at time k on the j th Monte Carlo realisation. The time-averaged RMS position error E is then calculated by

$$E = \sqrt{\frac{\sum_{j=1}^m \sum_{k=1}^l \sum_{i=1}^{t^k} (e_i^k(j))^2 \cdot \chi_{i,j,k}}{\sum_{j=1}^m \sum_{k=1}^l \sum_{i=1}^{t^k} \chi_{i,j,k}}} \quad (48)$$

To assess the performance of the algorithms, each Monte Carlo run has a different sequence of measurements but the j th Monte Carlo run for all the algorithms uses the same sequence [33]. The time-averaged RMS position error using non-thresholded data for all the algorithms is shown in Fig. 4

(a). JA performs more poorly because it does not perform subparticle crossover when the targets are far apart. Therefore, this PF acutely suffers from the curse of dimensionality that consists of a drop in performance when the dimension of the state increases [17]. The UKF does not perform properly either. It is better than all the PFs when there are only 100 particles but its performance is far from the performance of PP when the number of particles is high because the measurement model is highly non-linear. Nevertheless, as pointed out in [45], the UKF has much better performance than JA in a multiple target tracking scenario. CP performs better than IP as expected [6] but much worse than PP. This is due to the fact that IP and CP draw subparticles of the targets without taking into account the other targets. This is of crucial importance when we use non-thresholded data and there is a crossing in the targets' trajectories. Moreover, as AA does not perform subparticle crossover when the targets are in the same cluster, its performance is not as good as PP's because it is difficult to find particles in which all subparticles are good for representing the multitarget state when the number of particles is low. Nevertheless, when the number of particles increases, the importance of subparticle crossover decreases as it is easier to have particles in which all subparticles are appropriate for the multitarget state and the difference between AA and PP becomes smaller. On the other hand, MBRFS performance is far from PP, AA and CP performances. This was expected as MBRFS using non-thresholded data was developed under the assumption that the targets are far from each other [3]. We should clarify that we have not used the K-means algorithm for partition sorting in the PFs as suggested in [6]. K-means is an enforced method to speed up the removal of the multimodality of the posterior PDF after a target crossing [52] to facilitate target state estimation. Thus, including it would depart from the Bayesian perspective we want to follow in this paper. Nevertheless, we have also implemented it and, for this scenario, its effects are negligible as targets cross with different speeds so there are very few partition swappings [6].

The time-averaged RMS position error using binary data for all the algorithms is shown in Fig. 4 (b). The use of binary data implies a higher error as there is less information available. We can draw the same conclusions about the algorithms as when using non-thresholded data. However, as the number of particles increases, the difference between AA and PP is lower than before. This means that subparticle crossover is not so important with binary sensors. This can be attributed to the severe quantisation of the measurements making it is easier to find full particles that agree with the measurement.

Further insight into the behaviour of the various algorithms can be gained by plotting the RMS position error and the estimated number of targets against time. These plots are shown in Fig. 5 for non-thresholded data using 500 particles for the PFs. We divide the simulation into two intervals to explain the plots:

- First interval: Before time step 85. The four targets are far from each other: IP, CP, PP have the

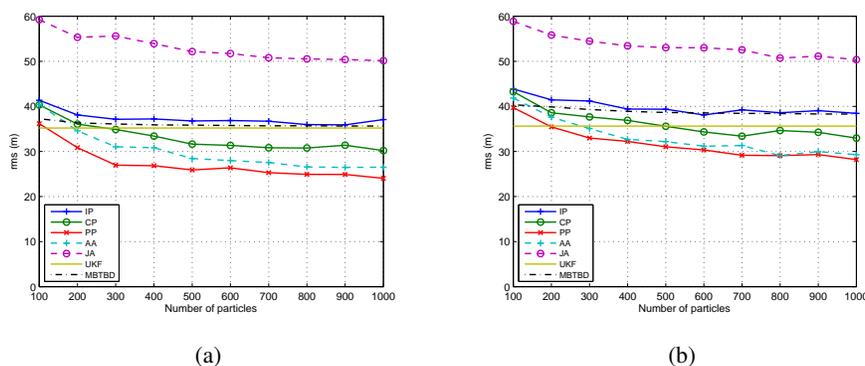
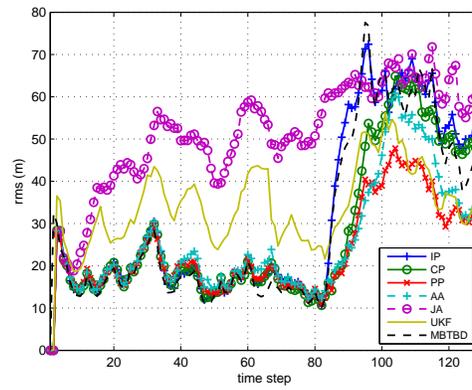


Figure 4 – Time-averaged RMS position error using non-thresholded data (a) and binary data (b). PP method outperforms the rest of the PFs because it performs subparticle crossover and considers nearby targets.

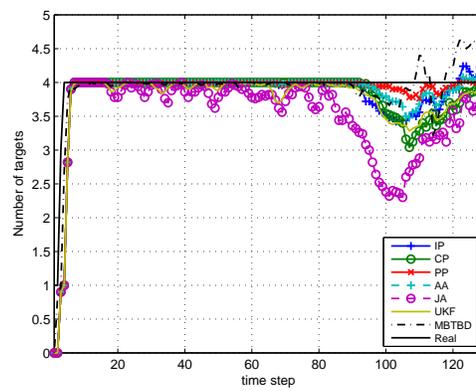
same performance as they are indeed the same PF when the targets are far from each other. AA has high performance as well because it also uses subparticle crossover when the targets are not in the same cluster. MBRFS also provides good estimates as the targets are not close. On the contrary, JA does not work properly because it does not use subparticle crossover. The UKF does not work properly either because the measurement model is highly nonlinear.

- Second interval: From time step 85 to 130. There are targets in close proximity. PP is the algorithm with the highest performance in RMS error and estimating the number of targets because it takes into account nearby targets and it also performs subparticle crossover. AA and UKF are roughly the algorithms with the second highest performance as they also take into account nearby targets. Conversely, IP and CP performances slump because they do not take into account targets in close proximity to propagate the particles. MBRFS does not perform well either because the targets are close to each other.

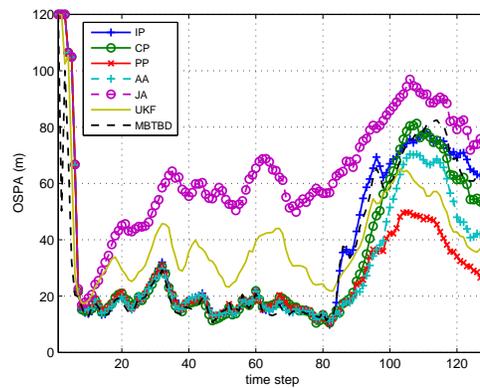
We have analysed the RMS error and the number of targets of the algorithms using different metrics to show how the algorithms perform for these two different aspects. However, the optimal subpattern assignment (OSPA) metric can assess both performances simultaneously [4]. Therefore, for the sake of completeness, we also show the mean OSPA (MOSPA) position error per time step in Figure 5 (c). The MOSPA error has been calculated with parameters $c = 120$ m and $p = 2$ [4]. The same conclusions can be drawn as with the previous metrics although MOSPA error does not indicate what part of the error corresponds to target number and to position error. At the beginning of the simulation, all the algorithms have a MOSPA error of $c = 120$ m as no targets have been added yet.



(a)



(b)



(c)

Figure 5 – RMS position error (a) and estimated number of targets (b) using non-thresholded data per time step with 500 particles (c) MOSPA position error. Before time step 85, IP, CP, PP, AA and MBRFS have roughly the same performance because the targets are far from each other. When the targets get closer, PP outperforms the rest.

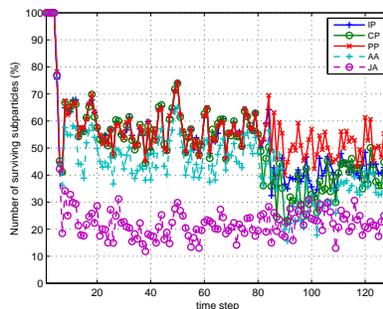


Figure 6 – Number of surviving subparticles using 500 particles. PP has the highest survival rate.

Table VI – Execution times in seconds for 500 particles

	IP	CP	PP	AA	JA	UKF	MBRFS
Non-thresholded	32.7	35	34.3	37.4	39.2	0.5	5.5
Binary	29.8	30.5	30.2	30.1	32.1	0.5	5.4

We also show the number of surviving subparticles per time step in Figure 6. We should note that the implemented PFs perform a resampling stage after each time step. Then, the subparticles at time k are based on only some subparticles at time $k - 1$ (the surviving subparticles) because the rest have not been selected because of their low weight. The higher the number of surviving particles, the most likely there are particles that represent the posterior well. A survival rate of between 30–80% is generally seen as acceptable [18]. Before time step 85, IP, CP, PP and AA have the same number of surviving subparticles. However, when the targets get closer, PP has the highest number of surviving subparticles and is able to maintain an acceptable survival rate. Before time step 4, all the algorithms have 100% survival rate as no targets have been added yet.

In Table VI, we show the execution times of the algorithms implemented in Matlab with C-MEX subroutines on a Pentium IV with a 2.4-GHz processor. The UKF is the fastest algorithm as it only approximates the first two moments of the posterior. In addition, processing non-thresholded data takes more time than processing binary data as the likelihood is faster to evaluate with binary data, see (17) and (19). For this scenario, computational time increases roughly linearly with the number of sensors and the number of particles.

Lastly, we show the number of particles that are needed to obtain a certain performance in a scenario with an increasing number of targets in Table VII. In the considered scenario, the number of targets whose trajectories cross at the same time is half the total number of targets and they do it in groups of

Table VII – Required number of particles to obtain an RMS position error lower than 25 m

Method	Number of targets							
	4	8	12	16	20	24	28	32
PP	300	300	300	500	500	800	800	900
MBRFS	8700	8700	8700	8700	8700	8700	8700	8700
IP	>10000	>10000	>10000	>10000	>10000	>10000	>10000	>10000
CP	600	8300	>10000	>10000	>10000	>10000	>10000	>10000
AA	400	1600	2800	6200	>10000	>10000	>10000	>10000

two targets. More details of this scenario are given in [9]. For a given error, MBRFS is able to track 32 targets with the same, albeit large, number of particles as with 4 targets because of its nonoverlapping assumption. This suggests that MBRFS would be the most suitable algorithm for tracking large numbers of targets. This conclusion is true only if targets remain well-separated. If several targets move in close proximity then the performance of MBRFS deteriorates, as seen in Figures 4 and 5. On the other hand, the proposed method (PP) provides a favourable trade-off between performance and computational expense. It is not limited to the case of well-separated targets and the required sample size increases by only three times as the number of targets is increased from 4 to 32. The remaining PFs fare much worse.

B. Estimation of the number of targets

Now, we compare our algorithm to the existence grid, whose details can be found in [7], [25], and the MBRFS with regard to the creation and deletion of tracks. The existence grid is a method to construct the importance densities governing target addition $q_2(\cdot)$ and removal $q_3(\cdot)$ based on partitioning the surveillance area into cells that form a rectangular grid. The pixelised sensor model is used in this section because the existence grid was designed to deal with this sensor model [7].

The existence grid provides a measure of the chances that a target is located in a cell, denoted as g_j^k in cell j at time k [7]. It is updated sequentially according to the measurement, the prior probability of adding a target in a cell α and the prior probability of removing a target β . In our simulations, rather than using the equation of the importance density to remove targets using the existence grid, equation (53) in [7], which permits a target removal probability exceeding one⁵, we use the following probability

⁵This happens for instance when there are two targets, $g_1^k = 0.99$, $g_2^k = 0.001$ and $\beta = 0.6$, then $\tau_2^k = 1.188$ which is not correct as it is a probability.

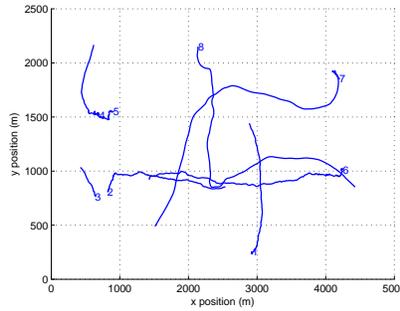


Figure 7 – Trajectories of the targets used to analyse the estimation of the number of targets

of removing target i :

$$\tau_i^k = \frac{\beta \cdot (1 - g_{v_i^k}^k)}{\sum_{l=1}^{t^{k-1}} (1 - g_{v_l^k}^k)} \quad (49)$$

where v_i^k denotes the cell occupied by the i th target at time k . In (49), the division of the denominator by the number of targets, which appears in (53) of [7], has been removed. The resulting target removal probability is at most β . The existence grid does not take into account targets' dynamics because the equation that models them, equation (11), is not used and, besides, it does not account for the possibility that a target might move to another cell. Then, the performance of the existence grid is much lower than the performance of our two-layer PF, especially when the targets move fast. In this case, performance mainly refers to the ability of the filter to create and delete tracks in a timely manner.

The scenario we use is shown in Fig. 7. The prior parameters are the same as in the previous example, see Table V, but with $\sigma_u = 10 \text{ m/s}^{3/2}$. The parameters of the two-layer PF are $N_b = 3$, $\eta_b = e^7$, $N_d = 5$, $\eta_d = e^{-2}$ for the two-layer PF. The parameters for the existence grid are $\alpha = 2 \cdot 10^{-15}$ and $\beta = 0.9$. The surveillance area is a square whose side is 5000 m long. This region is divided into 50×50 cells and each cell scans a $100 \text{ m} \times 100 \text{ m}$ area.

The estimated number of targets using non-thresholded data with 500 particles is shown in Fig. 8. Firstly, we should note that the existence grid does not work for targets that move fast. It properly creates tracks but fails to delete them. MBRFS has trouble detecting target 4 as it is in close proximity to target 5 at time 80 because it cannot deal with targets in close proximity. With low SNR , MBRFS does not detect target 4. MBRFS clearly overestimates the number of targets at around time 110 when targets 1 and 6 cross. The two-layer PF estimates the number of targets better than the other methods. Its main problem is that there is a slight delay in the creation and deletion of tracks but clearly controlled by N_b and N_d .

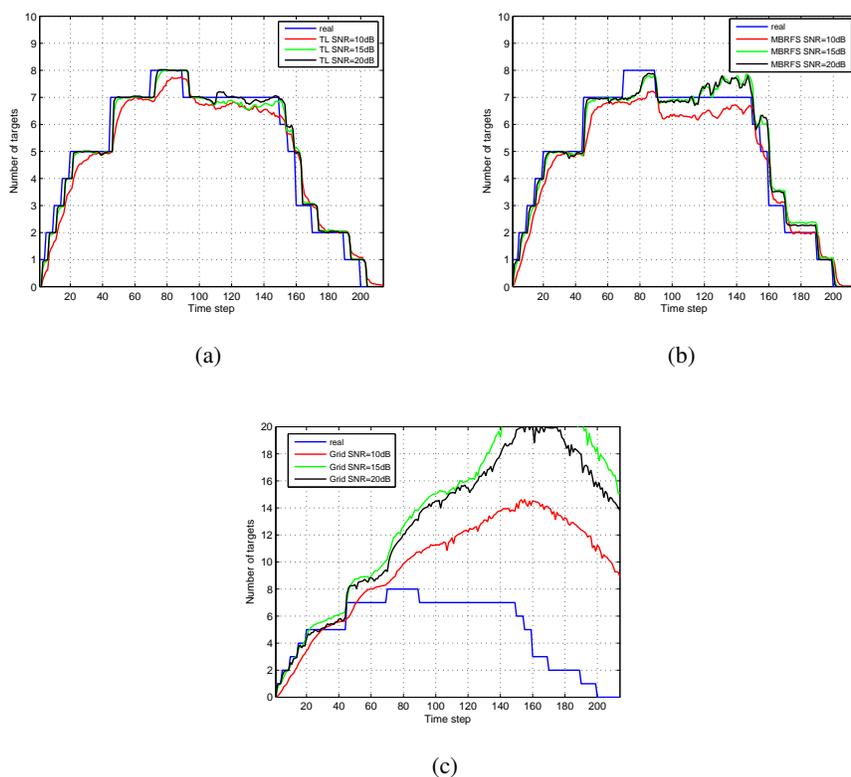


Figure 8 – Estimated number of targets: (a) Two-layer PF. (b) MBRFS. (c) Existence grid. The two-layer PF works better than the rest.

As the SNR increases, MBRFS and the two-layer PF are able to estimate the number of targets faster and more accurately. The main difference between the performances at 10 dB and 20 dB are the speeds with which tracks are initiated and deleted. The delayed track initiation at low SNR can be attributed to two factors. First, the creation of pre-tracks is often delayed because the probability of receiving a sufficiently strong measurement is lower at low SNR . Second, fewer pre-tracks are transferred to the Track PF because, at low SNR , tracking is more difficult and the likelihood ratio (42) tends to be lower. The difficulty of tracking and the decrease in the likelihood ratio also explain the slower track deletion seen at low SNR .

VI. CONCLUSIONS

We have developed a new method for the detection and tracking multiple targets adding unique labels to the target states. The labels allow the construction of tracks based on target estimates over time when the number of targets is unknown and changes with time. Importantly, the labelling of the tracks, which

is necessary to maintain the correspondence of target states over time, is done within the equations of the filter without any *ad hoc* methods. The use of the labels implies that, in principle, we are able to determine the probability that a target state estimate corresponds with a given target. This was done for a two-target case in [53] and [9, Section 5.3]. We have used two kinds of sensor models to assess the performance of the PP method and the two-layer PF. Nevertheless, the PP method and the two-layer PF are general algorithms and could be used for any measurement model, they do not rely on a specific sensor model. In addition, the novelties introduced by the PP method and the two-layer PF also apply in the RFS framework because of the equivalence between the posterior density of the labelled multitarget state in the vector and set formulations.

The PP method has been devised to sample the posterior PDF of the multitrack state of the surviving tracks. It has a low computational burden compared to other PF methods that aim to approximate the posterior of the multitrack state and outperforms a variety of algorithms previously proposed in the literature. The benefits of the PP method are more important if the number of targets in close proximity is high and the number of particles is low. As for sampling the number of targets, we have created a two-layer PF in which a subsidiary PF is used to build the importance density that samples the state of the new born tracks. This enables us to take target dynamics into account to initiate tracks. The current form of the two-layer PF creates and deletes tracks based on LRTs independently for each track. While this approach has a low computational burden, it could be useful to use multiple hypothesis tests on the whole multitarget state [54]. This would need much more computational resources but it is expected to improve detection performance in even more challenging scenarios. Besides, rather than using fixed time lags and thresholds to evaluate the LRTs, we could use variable time lags and thresholds using sequential detection [54].

APPENDIX

This appendix provides the proof of (23) under Assumption 1. As can be seen in (22), the Rao-Blackwellised PF approximation to the prior at time $k + 1$ is

$$p\left(\mathbf{P}^{k+1} \mid \mathbf{z}^{1:k}\right) = \sum_{i=1}^{N_{par}} w_i^k \prod_{j=1}^t p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,i}^k\right) \quad (50)$$

Under Assumption 1, the weights are evenly distributed, so (50) can be written as

$$p\left(\mathbf{P}^{k+1} \mid \mathbf{z}^{1:k}\right) = \sum_{i=1}^{N_{par}} \frac{1}{N_{par}} \prod_{j=1}^t p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,i}^k\right) \quad (51)$$

The marginal posterior density of the position of the j th target is obtained by integrating out the positions of the rest of the targets in (51)

$$p\left(\mathbf{p}_j^{k+1} \mid \mathbf{z}^{1:k}\right) = \sum_{i=1}^{N_{par}} \frac{1}{N_{par}} p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,i}^k\right) \quad (52)$$

Under Assumption 1, the tracks are a priori independent so the multitarget state prior PDF can be written as the product over the marginals:

$$p\left(\mathbf{P}^{k+1} \mid \mathbf{z}^{1:k}\right) = \prod_{j=1}^t \sum_{i=1}^{N_{par}} \frac{1}{N_{par}} p\left(\mathbf{p}_j^{k+1} \mid \mathbf{x}_{j,i}^k\right) \quad (53)$$

This completes the proof of (23).

REFERENCES

- [1] R. Mahler, ““Statistics 101” for multisensor, multitarget data fusion,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 53–64, Jan. 2004.
- [2] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.
- [3] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter, “Joint detection and estimation of multiple objects from image observations,” *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5129–5141, Oct. 2010.
- [4] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.
- [5] R. P. S. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
- [6] C. Kreucher, K. Kastella, and A. O. Hero III, “Multitarget tracking using the joint multitarget probability density,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.
- [7] M. R. Morelande, C. M. Kreucher, and K. Kastella, “A Bayesian approach to multiple target detection and tracking,” *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1589–1604, May. 2007.
- [8] C. M. Kreucher, A. O. Hero III, K. D. Kastella, and M. R. Morelande, “An information-based approach to sensor management in large dynamic networks,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 978–999, May 2007.
- [9] A. F. García-Fernández, “Detection and tracking of multiple targets using wireless sensor networks,” Ph.D. dissertation, Universidad Politécnica de Madrid, 2011. [Online]. Available: <http://oa.upm.es/9823/>
- [10] H. Blom and E. Bloem, “Permutation invariance in Bayesian estimation of two targets that maneuver in and out formation flight,” in *12th International Conference on Information Fusion*, July 2009, pp. 1296–1303.
- [11] H. Zhu, C. Han, and Y. Lin, “Particle labeling PHD filter for multi-target track-valued estimates,” in *Proceedings of the 14th International Conference on Information Fusion*, July 2011, pp. 1–8.
- [12] W.-K. Ma, B.-N. Vo, S. Singh, and A. Baddeley, “Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach,” *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3291–3304, Sept. 2006.
- [13] S. Mori, C.-Y. Chong, E. Tse, and R. Wishner, “Tracking and classifying multiple targets without a priori identification,” *IEEE Transactions on Automatic Control*, vol. 31, no. 5, pp. 401–409, May 1986.

- [14] P. J. Green, "Trans-dimensional Markov chain Monte Carlo," in *Highly Structured Stochastic Systems*, P. J. Green, N. L. Hjort, and S. Richardson, Eds. Oxford University Press, 2003.
- [15] —, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, Dec. 1995.
- [16] C. Andrieu and A. Doucet, "Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2667–2676, Oct. 1999.
- [17] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proceedings IEEE Aerospace Conference*, vol. 4, 8-15 2003, pp. 1979–1993.
- [18] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216–223, Feb. 2002.
- [19] C. M. Kreucher, M. R. Morelande, K. Kastella, and A. O. Hero III, "Joint multi-target particle filtering," in *Foundations and Applications of Sensor Management*, A. O. Hero III, D. A. Castañón, D. Cochran, and K. Kastella, Eds. Springer Science + Business Media, 2008.
- [20] V. Cevher and J. H. McClellan, "General direction-of-arrival tracking with acoustic nodes," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 1–12, Jan. 2005.
- [21] I. Kyriakides, D. Morrell, and A. Papandreou-Suppappola, "Sequential Monte Carlo methods for tracking multiple targets with deterministic and stochastic constraints," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 937–948, March 2008.
- [22] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [23] J. Vermaak, S. J. Godsill, and P. Perez, "Monte Carlo filtering for multi target tracking and data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 309–332, Jan. 2005.
- [24] Y. Boers and J. N. Driessen, "Multitarget particle filter track before detect application," *IEE Proceedings Radar, Sonar and Navigation*, vol. 151, no. 6, pp. 351–357, Dec. 2004.
- [25] M. Fallon and S. Godsill, "Multi target acoustic source tracking with an unknown and time varying number of targets," in *Hands-Free Speech Communication and Microphone Arrays, HSCMA 2008*, May 2008, pp. 77–80.
- [26] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [27] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1237–1241, Jun. 1994.
- [28] A. F. García-Fernández and J. Grajal, "Multitarget tracking using the joint multitrack probability density," in *12th International Conference on Information Fusion*, July 2009, pp. 595–602.
- [29] S. Mori and C.-Y. Chong, "Point process formalism for multiple target tracking," in *Proceedings of the Fifth International Conference on Information Fusion*, vol. 1, 2002, pp. 10–17.
- [30] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [31] L. Kleinrock, *Queueing Systems*. John Wiley & Sons, 1976.
- [32] M. Vihola, "Rao-Blackwellised particle filtering in random set multitarget tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 689–705, April 2007.
- [33] Y. Bar-Shalom, T. Kirubarajan, and X. R. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

- [34] S. J. Davey, M. G. Rutten, and B. Cheung, "A comparison of detection performance for several track-before-detect algorithms," in *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008, pp. 1–10.
- [35] R. Niu and P. K. Varshney, "Performance analysis of distributed detection in a random sensor field," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 339–349, Jan. 2008.
- [36] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part I: Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, Jul. 1997.
- [37] W. D. Blair and M. Brandt-Pearce, "Unresolved Rayleigh target detection using monopulse measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 2, pp. 543–552, Apr. 1998.
- [38] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *Journal of the Royal Statistical Society: Series B*, vol. 64, pp. 827–836, 2002.
- [39] 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, Jun. 1999.
- [40] D. B. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, 1987.
- [41] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc., 2008.
- [42] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, Jul. 1983.
- [43] W. Yi, M. R. Morelande, L.-J. Kong, and J.-Y. Yang, "An efficient particle filter for multi-target tracking using an independence assumption," in *15th International Conference on Information Fusion*, July 2012, pp. 2378–2385.
- [44] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice-Hall, 1998.
- [45] M. R. Morelande, "Tracking multiple targets with a sensor network," in *9th International Conference on Information Fusion*, July 2006, pp. 1–7.
- [46] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [47] T. Vercauteren, D. Guo, and X. Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 714–723, April 2005.
- [48] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-Blackwellized particle filter for multiple target tracking," *Information Fusion*, vol. 8, no. 1, pp. 2–15, Jan. 2007.
- [49] J. Vermaak, S. Maskell, and M. Briers, "A unifying framework for multi-target tracking and existence," in *8th International Conference on Information Fusion*, July 2005, pp. 250–258.
- [50] C. Hue, J. Le Cadre, and P. Perez, "Tracking multiple objects with particle filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 791–812, Jul 2002.
- [51] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 481–497, March 2009.
- [52] Y. Boers, E. Sviestins, and H. Driessen, "Mixed labelling in multitarget particle filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 792–802, April 2010.
- [53] A. F. García-Fernández, M. R. Morelande, and J. Grajal, "Particle filter for extracting target label information when targets move in close proximity," in *14th International Conference on Information Fusion*, 2011, pp. 795–802.

- [54] A. F. García-Fernández and J. Grajal, "Sequential multiple target detection using particle filters," in *IEEE Workshop on Statistical Signal Processing*, 2011, pp. 749–752.



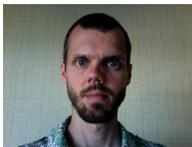
Ángel F. García-Fernández received his degree in Telecommunication Engineering (with honours) and the Ph.D. from Universidad Politécnica de Madrid, Spain, in 2007 and 2011, respectively.

Since 2007 he has been a researcher at the Department of Signals, Systems and Radiocommunications of the same university. His research activities and interests are in the area of Bayesian filtering, multitarget tracking, wireless sensor networks, and radar signal processing.



Jesús Grajal was born in Toral de los Guzmanes (León), Spain, in 1967. He received the Ingeniero de Telecomunicación and the Ph.D. degrees from the Technical University of Madrid, Madrid, Spain in 1992 and 1998, respectively.

Since 2001 he has been an Associate Professor at the Signals, Systems, and Radiocommunications Department of the Technical School of Telecommunication Engineering of the same University. His research activities are in the area of hardware-design for radar systems, radar signal processing and broadband digital receivers for radar and spectrum surveillance applications.



Mark R. Morelande received the B.Eng. degree in aerospace avionics from Queensland University of Technology, Brisbane, Australia in 1997 and the Ph.D. in electrical engineering from Curtin University of Technology, Perth, Australia in 2001.

In 2001 he was a Postdoctoral Fellow at the Centre for Eye Research, Queensland University of Technology. From 2002-2005 he was a Research Fellow at the Cooperative Research Centre for Sensor, Signal and Information Processing, University of Melbourne. He is now a Senior Research Fellow in the Melbourne Systems Laboratory, also at The University of Melbourne. His research interests include non-stationary signal analysis and target tracking with particular emphasis on multiple target tracking and sequential Monte Carlo methods.