PREPRINT: Annual Conf. Inf. Fusion, Salamanca, Spain, July 2014

MCMC-based posterior independence approximation for RFS multitarget particle filters

Ángel F. García-Fernández, Ba-Ngu Vo, Ba-Tuong Vo
Department of Electrical and Computer Engineering, Curtin University, Australia
Emails: {angel.garciafernandez, ba-ngu.vo, ba-tuong.vo}@curtin.edu.au

***Abstract*—The objective of this paper is to approximate the unlabelled posterior random finite set (RFS) density in multitarget tracking (MTT) using particle filters (PFs). The unlabelled posterior can be equivalently represented by any labelled density that belongs to the posterior RFS family. For the limited number of particles used in practice, PFs that assume posterior independence among target states outperform those without it. Consequently, we can improve the PF approximation by aiming at the labelled density within the posterior RFS family whose target states are as independent as possible. In this paper, we focus on the case of fixed and known number of targets and propose an algorithm based on Markov chain Monte Carlo (MCMC) that pursues this aim. This algorithm can be added to any PF with posterior independence assumption.**

***Index Terms*—MCMC, particle filter, multitarget tracking**

I. Introduction

In the traditional random finite set (RFS) formulation to multiple target tracking (MTT), we are interested in estimating the number of targets along with their states based on a sequence of measurements [1]. In this case, all the information of interest about the targets is contained in the unlabelled posterior (RFS) probability density function (PDF). In theory, this density can be calculated recursively using the prediction and update steps.

If the target states are augmented with a unique label as in [2], [3], the prediction and update steps are more easily performed because of the simplification of the set integrals in the labelled case. Nevertheless, the unlabelled posterior can be represented by any labelled posterior that belongs to the posterior RFS family [4], which represents the set of labelled PDFs that correspond to the same unlabelled PDF by integrating out the labels. In addition, the subsequent applications of the prediction and update steps to any such labelled posterior also gives a labelled posterior that belongs to the posterior RFS family [4]. Therefore, even in the unlabelled case, it is useful to perform filtering using a labelled posterior PDF. Importantly, in the unlabelled case, we have the advantage that at any moment we can choose another labelled posterior PDF within the posterior RFS family at our convenience [4].

In this paper, we set aside the problem of estimating the target number and focus on the case of fixed and known number of targets. In most cases of interest, calculating the posterior (either labelled or unlabelled) is intractable due to nonlinear/non-Gaussian dynamic or measurement models. Consequently, it must be approximated, for instance, using a particle filter (PF) [5]. In practice, to ensure a low computational burden, PFs are implemented with a reasonably low number of particles. In this case, posterior independence assumption PFs (PIA-PFs), which assume posterior independence among target states, outperform those without this assumption [3], [6], [7]. One explanation behind this improvement is that PIA-PFs provide a more accurate approximation to the prior at the next time step under Gaussian assumptions [6]. Evidently, the more independent the targets are, better performance is expected.

Based on the above mentioned insights, the aim of this paper is to find the PIA-PF approximation to the labelled PDF of the posterior RFS family in which target states are most independent. This is of high importance as this PIA-PF approximation represents the unlabelled posterior more accurately than any other PIA-PF approximation. To do so, we first develop a recursion which decreases the Kullback-Leibler divergence (KLD) of a labelled posterior within the RFS family from a labelled posterior with independent target states. This recursion is a generalised version of the algorithm found in [4], which was developed for two targets with the objective of finding the best independent Gaussian approximation. The practical implementation of the recursion is performed based on Markov chain Monte Carlo (MCMC) steps after the resampling step. We demonstrate the power of this idea in an MTT example in which targets get in close proximity.

The remainder of the paper is organised as follows. The problem is formulated in Section II. The recursion that lowers the KLD of a labelled posterior within the RFS family from a labelled posterior with independent target states is provided in Section III. The MCMC algorithm that approximates this recursion is given in Section IV. In Section V, the performance of the algorithm is analysed in a numerical example. Finally, conclusions are drawn in Section VI.

II. Problem formulation

In this paper, we make the following assumption

- A1 There is a fixed and known number t of targets, without any target births or deaths.

The collection of targets at time step k is represented by the unlabelled set $X^k = \{\mathbf{x}_1^k, ..., \mathbf{x}_t^k\}$, where $\mathbf{x}_i^k \in \mathbb{R}^{n_x}$ is the state of the ith target at time k. The ith target evolves with a transition density $g\left(\cdot \left| \mathbf{x}_i^k \right.\right)$. At time step k, targets are observed through noisy measurements. These measurements

can be either a vector $\mathbf{z}^k \in \mathbb{R}^{n_{z,1}}$, as in sensor networks, or a set $Z^k \subset \mathbb{R}^{n_{z,2}}$, as in point detection measurement models used in radar. This paper is general and applies to both measurement models. We assume that measurements do not provide any labelling information, therefore, once we observe the kth measurement, the multi-target likelihood $\ell^k(\cdot)$ depends on X^k, as it is permutation invariant. For the sake of notational simplicity, we omit the explicit value of the measurement in the likelihood.

The objective is to calculate the unlabelled RFS posterior density $\breve{\pi}^k(\cdot)$, whose argument is X^k. Under Assumption A1, the filtering recursion for an unlabelled set state is

$$\breve{\varpi}^k\left(X^k\right) = \int \prod_{j=1}^{t} g\left(\mathbf{x}_j^k \left| \mathbf{x}_j^{k-1}\right.\right) \breve{\pi}^{k-1}\left(X^{k-1}\right) \delta X^{k-1} \quad (1)$$

$$\breve{\pi}^k\left(X^k\right) \propto \ell^k\left(X^k\right) \breve{\varpi}^k\left(X^k\right) \quad (2)$$

where $\propto$ denotes proportionality and $\breve{\varpi}^k(\cdot)$ is the RFS prior.

A. Calculation of the unlabelled RFS posterior using a labelled posterior

In this section, we first summarise some key results in [4] that explain how an unlabelled RFS posterior can be calculated using a labelled posterior. Second, we sketch how we use this idea in our paper.

The main benefit of calculating the unlabelled RFS posterior via a labelled RFS posterior is the simplification of the set integrals for labelled sets [2]. As we use A1, the labelled set state can be represented as the multitarget state vector $\mathbf{X}^k = \left[\left(\mathbf{x}_1^k\right)^T, \left(\mathbf{x}_2^k\right)^T, ..., \left(\mathbf{x}_t^k\right)^T\right]^T \in \mathbb{R}^{tn_x}$, where T stands for transpose [4], [8]. In addition, under A1, an unlabelled (RFS) PDF is referred to as RFS PDF while a labelled (RFS) PDF is simply referred to as PDF . The filtering recursion for the multitarget state vector is

$$\varpi^k\left(\mathbf{X}^k\right) = \int \prod_{j=1}^{t} g\left(\mathbf{x}_j^k \left| \mathbf{x}_j^{k-1}\right.\right) \pi^{k-1}\left(\mathbf{X}^{k-1}\right) d\mathbf{X}^{k-1} \quad (3)$$

$$\pi^k\left(\mathbf{X}^k\right) \propto \ell^k\left(\mathbf{X}^k\right) \varpi^k\left(\mathbf{X}^k\right) \quad (4)$$

where $\pi^k(\cdot)$ is the posterior, $\varpi^k(\cdot)$ is the prior and the integral is a vector integral.

Definition 1. A PDF $\pi^k(\cdot)$ belongs to the RFS family[1] $\mathcal{R}_{\breve{\pi}^k}$ of $\breve{\pi}^k(\cdot)$ if and only if [4]:

$$\breve{\pi}^k\left(X^k\right) = \sum_{p=1}^{t!} \pi^k\left(\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right)\right) \quad (5)$$

where vectors $\boldsymbol{\phi}_p = [\phi_{p,1}, ..., \phi_{p,t}]^T$ $p \in \{1, ..., t!\}$ represent the permutations of vector $[1, ..., t]^T$ and $\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X}) = \left[\left(\mathbf{x}_{\phi_{p,1}}\right)^T, \left(\mathbf{x}_{\phi_{p,2}}\right)^T, ..., \left(\mathbf{x}_{\phi_{p,t}}\right)^T\right]^T$ is the permutation of $\mathbf{X}^k$ indicated by $\boldsymbol{\phi}_p$.

The following property is met [4]

$$\pi^{k-1}(\cdot) \in \mathcal{R}_{\breve{\pi}^{k-1}} \Rightarrow \pi^k(\cdot) \in \mathcal{R}_{\breve{\pi}^k}. \quad (6)$$

[1]In the theory of point processes, $\mathcal{R}_{\breve{\pi}^k}$ represents the PDFs that are countably equivalent to $\breve{\pi}^k(\cdot)$ [9].

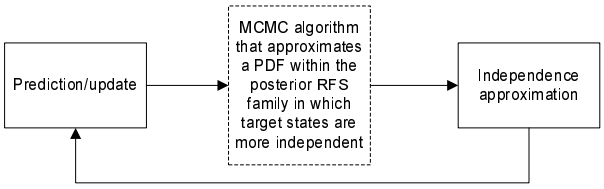


Figure 1: Diagram of an PIA-PF recursion. An MCMC-based algorithm is developed in this paper to improve the posterior independence approximation within the RFS posterior family.

That is, if a PDF $\pi^{k-1}(\cdot)$ at time $k-1$ belongs to $\mathcal{R}_{\breve{\pi}^{k-1}}$, the PDF $\pi^k(\cdot)$ at time k, which is obtained using (3) and (4), belongs to $\mathcal{R}_{\breve{\pi}^k}$. From a PDF we can obtain its corresponding RFS PDF using (5). Therefore, instead of using (1) and (2) to calculate $\breve{\pi}^k(\cdot)$, we can use a PDF that belongs to $\mathcal{R}_{\breve{\pi}^k}$ and perform filtering using (3) and (4), which are simpler to calculate.

In general, equations (3) and (4) cannot be calculated so they must be approximated. Importantly, as pointed out in [4], at any time step, $\pi^k(\cdot)$ can be changed by any other PDF than belongs to the same RFS family. This is quite useful because there are some PDFs within the same RFS family that are more easily approximated than others. For example, this fact was used in [4] to provide more accurate Gaussian approximations.

In this paper, we use the fact that, in PF approximations with a low number of particles, it is advantageous to assume independence among target states [6]. The RFS family does not have to contain a PDF in which the targets are independent. Nevertheless, ideally, we would like to use the PDF $\nu^k(\cdot)$ with independent targets that most closely resembles a member of the RFS family for the next prediction and update steps. In the next section, we derive a recursion that provides a sequence of PDFs within the RFS familty that improves this target independence approximation. This will be used to design an algorithm based on MCMC that is expected to improve the performance of PIA-PFs in Section IV. The resulting PF recursion is illustrated in Figure 1.

III. Improvement of the independence approximation

In this section we provide a recursion to select a PDF that belongs to $\mathcal{R}_{\breve{\pi}^k}$ in which target states are more independent than in $\pi^k(\cdot)$. We assess how independent the target states are using the KLD. First, we find it convenient to write the family $\mathcal{R}_{\breve{\pi}^k}$ in terms of a function $\alpha(\cdot)$. As shown in Appendix A

$$\pi^k(\cdot) \in \mathcal{R}_{\breve{\pi}^k} \Leftrightarrow \exists \alpha(\cdot) : \pi^k\left(\mathbf{X}^k\right) = \alpha\left(\mathbf{X}^k\right) \breve{\pi}^k\left(\left\{\mathbf{x}_1^k, ..., \mathbf{x}_t^k\right\}\right) \quad (7)$$

with

$$\sum_{p=1}^{t!} \alpha\left(\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right)\right) = 1 \quad (8)$$

and $\alpha\left(\mathbf{X}^k\right) \geq 0$. Therefore, by multiplying the RFS density by $\alpha(\cdot)$, we get a labelled density that belongs to $\mathcal{R}_{\breve{\pi}^k}$.

A widely used way to measure the similarity between PDFs is the KLD. The aim is to choose the PDF that belongs to

$\mathcal{R}_{\breve{\pi}^k}$ which is more closely approximated by a PDF with independent target states. Thus, we want to find

$$\nu^{k,\star}(\cdot) = \underset{\nu^k(\cdot)}{\arg\min}\, \mathrm{D}\left(\pi^k(\cdot) \left\| \nu^k(\cdot)\right.\right)$$

with the constraints $\pi^k \in \mathcal{R}_{\breve{\pi}^k}$ and

$$\nu^k\left(\mathbf{X}^k\right) = \nu_1^k\left(\mathbf{x}_1^k\right)\nu_2^k\left(\mathbf{x}_2^k\right)...\nu_t^k\left(\mathbf{x}_t^k\right) \tag{9}$$

where $\mathrm{D}(\cdot\|\cdot)$ represents the KLD, $\nu^k(\cdot)$ is a PDF with independent target states and $\nu_i^k(\cdot)$ is the corresponding marginal PDF of the ith target.

In the rest of this section we derive a recursion that provides a sequence of PDFs $\pi^{k,0}(\cdot), \nu^{k,0}(\cdot), \pi^{k,1}(\cdot), \nu^{k,1}(\cdot), ...$ with decreasing KLD. The PDF $\pi^{k,i}(\cdot) \in \mathcal{R}_{\breve{\pi}^k}$ so we find it convenient to characterise it by $\alpha^i(\cdot)$ using (7).

We start the recursion with a density $\pi^{k,0}(\cdot)$. First, we find $\nu^{k,0}(\cdot)$ that minimises $\mathrm{D}\left(\pi^{k,0}(\cdot)\left\|\nu^{k,0}(\cdot)\right.\right)$. Then, we find $\alpha^1(\cdot)$ that minimises $\mathrm{D}\left(\alpha^1(\cdot)\breve{\pi}^k(\{\cdot\})\left\|\nu^{k,0}(\cdot)\right.\right)$, which is equivalent to finding $\pi^{k,1}(\cdot) \in \mathcal{R}_{\breve{\pi}^k}$ that minimises $\mathrm{D}\left(\pi^{k,1}(\cdot)\left\|\nu^{k,0}(\cdot)\right.\right)$. At every iteration the KLD gets smaller and smaller until convergence [4]. The following theorems indicate how the consecutive minimisations are performed.

Theorem 2. *For a given $\pi^k(\cdot)$, the solution to*

$$\underset{\nu^k(\cdot)}{\arg\min}\, \mathrm{D}\left(\pi^k(\cdot)\left\|\nu^k(\cdot)\right.\right)$$

with constraint (9) is given by marginalisation

$$\nu_i^k\left(\mathbf{x}_i^k\right) = \int \pi^k\left(\mathbf{X}^k\right) d\mathbf{x}_1^k ... d\mathbf{x}_{i-1}^k d\mathbf{x}_{i+1}^k ... d\mathbf{x}_t^k \quad i = \{1, ..., t\}. \tag{10}$$

Theorem 2 is proved in Appendix B.

Theorem 3. *For a given $\nu^k(\cdot)$, the solution to*

$$\underset{\alpha(\cdot)}{\arg\min}\, \mathrm{D}\left(\alpha(\cdot)\breve{\pi}^k(\{\cdot\})\left\|\nu^k(\cdot)\right.\right)$$

with constraint (8) is given by

$$\alpha\left(\mathbf{X}^k\right) = \frac{\nu^k\left(\mathbf{X}^k\right)}{\sum_{p=1}^{t!}\nu^k\left(\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right)\right)}. \tag{11}$$

Theorem 3 is proved in Appendix C. Note that constraint (8) means that we are looking for the member of $\mathcal{R}_{\breve{\pi}^k}$ which more closely represents the PDF $\nu^k(\cdot)$. The algorithm foundations are the same of the algorithm in [4] although there are some differences. In [4], the authors propose a recursion to find the best Gaussian approximation in a two-target case and then use moment matching to obtain an independent Gaussian posterior approximation. In our case, we directly look for an independent approximation, we are not restricted to the family of Gaussian PDFs and the derivation applies to any number of targets.

A. Illustrative example

In this subsection, we provide an example to illustrate the above mentioned recursion. We assume there are two one-dimensional targets and the initial PDF is

$$\pi^{k,0}\left(\mathbf{X}^k\right) = \mathcal{N}\left(\mathbf{X}^k; [10, 11]^T, \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}\right) \tag{12}$$

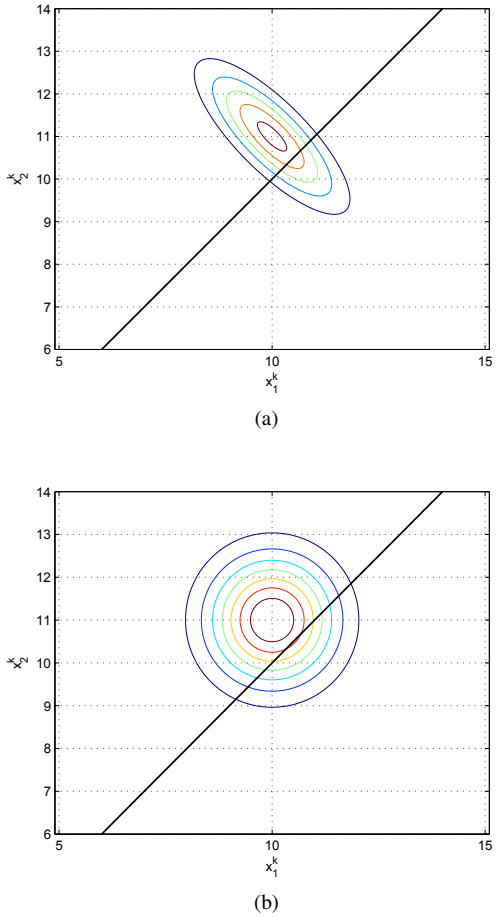


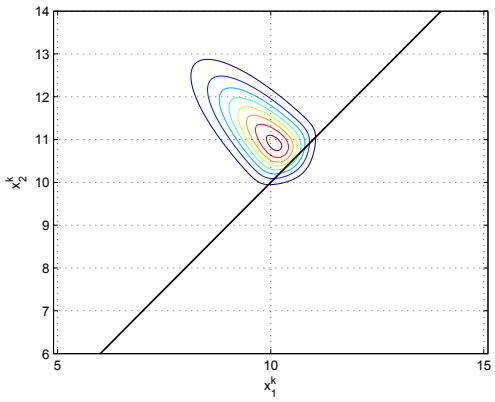
Figure 2: Contour plots of (a) the initial PDF $\pi^{k,0}(\cdot)$ and (b) its best approximation $\nu^{k,0}(\cdot)$ with independent targets. The line $\mathbf{x}_2^k = \mathbf{x}_1^k$ is also plotted.

where $\mathcal{N}\left(\mathbf{X}^k; \overline{\mathbf{X}}^k, \boldsymbol{\Sigma}\right)$ is the Gaussian PDF with mean $\overline{\mathbf{X}}^k$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\mathbf{X}^k$, $\sigma_1 = \sigma_2 = 1$ and $\rho = -0.8$. In this example, all the calculations are based on a grid of size 0.01×0.01. The best approximation $\nu^{k,0}(\cdot)$ to $\pi^{k,0}(\cdot)$ with independent targets is given by Theorem 2. Both PDFs are represented in Figure 2. The KLD between these two PDFs is 0.51. Now, we perform the recursion to obtain a PDF with independent target that represents more closely a PDF that belongs to the RFS family of $\pi^{k,0}(\cdot)$.

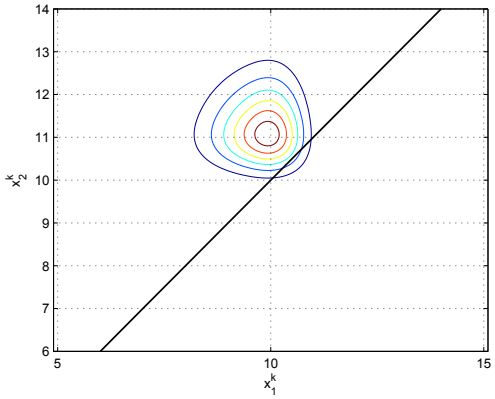
We perform six steps of the recursion and the KLD of $\nu^{k,i}(\cdot)$ from $\pi^{k,i}(\cdot)$ is given in Table I. In a few iterations, the KLD is around one half of the initial KLD. The PDFs at iteration 6 are plotted in Figure 3. The PDFs tend to concentrate in the region where $\mathbf{x}_2^k > \mathbf{x}_1^k$. It should be noted that this region is important in estimation based on the mean square optimal subpattern assignment (MSOSPA) error [10], [11]. We recall that the importance of this recursion is due to the fact that a PF with posterior independence assumption among target states based on $\pi^{k,6}(\cdot)$ and $\nu^{k,6}(\cdot)$ is expected to outperform another one based on $\pi^{k,0}(\cdot)$ and $\nu^{k,0}(\cdot)$ as the similarity between the former PDFs is higher.

Table I: KLD of $\nu^{k,i}(\cdot)$ from $\pi^{k,i}(\cdot)$

Iteration number i	0	1	2	3	4	5	6
KLD	0.51	0.41	0.32	0.27	0.26	0.26	0.26



(a)



(b)

Figure 3: Contour plots of (a) $\pi^{k,6}(\cdot)$ and (b) its best approximation $\nu^{k,6}(\cdot)$ with independent targets. These two PDFs are more similar than the initial ones, see Figure 2, based on the KLD.

IV. MCMC ALGORITHM FOR PIA-PFS

In this section, first, we review what the posterior looks like in a PIA-PF in Section IV-A. In Section IV-B, we provide an algorithm for PIA-PFs based on MCMC to approximate the recursion in Section III.

A. PIA-Particle filters

Let us assume the posterior PDF at time $k-1$ is given by a PF approximation with N particles and even weights. We denote the ith particle as $\mathbf{X}_i^{k-1} = \left[\left(\mathbf{x}_{1,i}^{k-1}\right)^T, ..., \left(\mathbf{x}_{t,i}^{k-1}\right)^T\right]^T$. In this case, using (3) and (4), the posterior PDF at time k becomes

$$\pi^k\left(\mathbf{X}^k\right) \propto \ell^k\left(\mathbf{X}^k\right) \sum_{i=1}^{N} \prod_{j=1}^{t} g_{j,i}\left(\mathbf{x}_j^k\right) \tag{13}$$

where $g_{j,i}\left(\mathbf{x}_j^k\right) = g\left(\mathbf{x}_j^k \left| \mathbf{x}_{j,i}^{k-1}\right.\right)$.

The objective is to obtain samples from (13). Directly sampling from (13) is intractable in general. A widely used solution is to use importance sampling, i.e., a PF, to obtain samples from it. We are interested in PF algorithms that are computationally efficient, i.e., which use a low number of particles. However, due to the high dimensionality of the problem, which increases with the number of targets, obtaining a low number of particles that represents (13) accurately enough is a challenging task. One way to tackle the problem is to make the assumption

- A2 The target states at time $k-1$ are independent

A2 is referred to as posterior independence assumption (PIA) in [6] where it is shown that it improves the PF approximation to the prior $\varpi^k(\cdot)$ at time k for low number of particles. This implies that PFs that have been designed under A1 outperform PFs without it for a low enough number of particles [3], [6], [7]. Under A1, the posterior at time k becomes [3]

$$\pi^k\left(\mathbf{X}^k\right) \propto \ell^k\left(\mathbf{X}^k\right) \prod_{j=1}^{t} \sum_{i=1}^{N} g_{j,i}\left(\mathbf{x}_j^k\right). \tag{14}$$

We can write (14) as [3]

$$\pi^k\left(\mathbf{X}^k, \mathbf{a}\right) \propto \ell^k\left(\mathbf{X}^k\right) \prod_{j=1}^{t} g_{j,a_j}\left(\mathbf{x}_j^k\right) \tag{15}$$

where each component of auxiliary vector $\mathbf{a} = [a_1, ..., a_t]^T \in \{1, ..., N\}^t$ is an index on the jth target mixture in (14). That is, for the ith multitarget particle $\left(\mathbf{X}_i^k, \mathbf{a}_i\right)$ at time k, $\mathbf{a}_i$ indicates the parents at time $k-1$ of the targets of the particle. The auxiliary vector $\mathbf{a}$ is quite useful as it lowers the computational burden and allows subparticle crossover [3].

Although the PIA implies a significant improvement in performance, a PF approximation to (15) with low number of particles can have a lack of sample diversity, which lowers performance. In order to improve sample diversity, MCMC steps can be applied to the particles after the resampling step [5]. The procedure is the following: first we obtain N weighted particles $\left(\mathbf{X}_i^k, \mathbf{a}_i\right)$ using a PF, second, we apply a resampling stage, and third, we apply MCMC steps.

In this paper, rather than using MCMC just to improve sample diversity, we use MCMC to approximate the sequence of PDFs explained in Section III. Not only does this improve sample diversity but it also improves the accuracy of the posterior independence approximation, which is required at the following time step due to Assumption A2. That is, at each step of the iteration of Section III, we obtain N samples from a PDF with independent targets that resembles more closely a PDF within the posterior RFS family of (15) than the PDF with independent targets at the previous step.

In the following, we assume we have N evenly weighted particles $\left(\mathbf{X}_i^k, \mathbf{a}_i\right)$ distributed according to (15). The MCMC steps we develop in the rest of the section are independent of how these samples have been obtained. For instance, we can use the parallel partition (PP) method [3] or the independent joint optimal importance density (IJOID) [6]. The evenly distributed particles can be obtained by performing resampling on weighted samples.

B. MCMC steps to improve the independence approximation

As mentioned in Section II, instead of approximating (15), we can approximate any labelled posterior that belongs to the same RFS family. The idea is to approximate the sequence of PDFs $\pi^{k,0}(\cdot), \nu^{k,0}(\cdot), \pi^{k,1}(\cdot), \nu^{k,1}(\cdot), ...$ indicated in Section III based on MCMC. This way, at every step of this sequence, we improve the accuracy of the PIA-PF posterior approximation. The output of this sequence is the input for the next prediction and update steps, as required by Assumption A2 and shown in Figure 1. This algorithm is referred to as independent target MCMC (IT-MCMC).

In this case, $\pi^{k}(\cdot)$ is given by (14) and $\pi^{k,l}(\cdot) \in \mathcal{R}_{\tilde{\pi}^k}$ so using (5) and (7), $\pi^{k,l}(\cdot)$ can be written as

$$\pi^{k,l}\left(\mathbf{X}^k\right) \propto \alpha^{l}\left(\mathbf{X}^k\right) \ell^{k}\left(\mathbf{X}^k\right) \sum_{p=1}^{t!} \prod_{j=1}^{t} \sum_{i=1}^{N} g_{j,i}\left(\mathbf{x}_{\phi_{p,j}}^{k}\right)$$

where $\alpha^{l}(\cdot)$ meets (8) and characterises $\pi^{k,l}(\cdot)$. It should be noted that $\pi^{k,l}(\cdot)$ is equivalent to applying Bayes' rule to the RFS prior with a modified version of the likelihood.

Using the auxiliary vector $\mathbf{a}$ as in (15), we get

$$\pi^{k,l}\left(\mathbf{X}^k, \mathbf{a}\right) \propto \alpha^{l}\left(\mathbf{X}^k\right) \ell^{k}\left(\mathbf{X}^k\right) \sum_{p=1}^{t!} \prod_{j=1}^{t} g_{j,a_j}\left(\mathbf{x}_{\phi_{p,j}}^{k}\right). \quad (16)$$

The recursion starts with the particles $\left(\mathbf{X}_i^{k,0}, \mathbf{a}_i\right)$ $i = 1, ..., N$ that provide a PF approximation to $\pi^{k,0}(\cdot) = \pi^{k}(\cdot)$. It can be performed if, given a PF approximation $\left(\mathbf{X}_i^{k,l}, \mathbf{a}_i\right)$ $i = 1, ..., N$ to $\pi^{k,l}(\cdot)$, we can obtain a PF approximation to $\pi^{k,l+1}(\cdot)$. In order to get samples from $\pi^{k,l+1}(\cdot)$, we need to find $\alpha^{l+1}(\cdot)$ using Theorems 2 and 3. First we indicate how to calculate $\alpha^{l+1}(\cdot)$ based on the PF approximation to $\pi^{k,l}(\cdot)$. Second, we explain an MCMC algorithm to obtain samples from $\pi^{k,l+1}(\cdot)$ once $\alpha^{l+1}(\cdot)$ is obtained.

1) Calculation of $\alpha^{l+1}(\cdot)$: In order to be able to run the MCMC algorithm that gives samples from $\pi^{k,l+1}(\cdot)$, we need to calculate $\alpha^{l+1}(\cdot)$ based on the PF approximation to $\pi^{k,l}(\cdot)$ and Theorems 2 and 3. The PF approximation to $\pi^{k,l}(\cdot)$ can be written as

$$\pi^{k,l}\left(\mathbf{X}^k\right) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(\mathbf{X}^k - \mathbf{X}_i^{k,l}\right) \quad (17)$$

where $\delta(\cdot)$ denotes the Dirac delta. Based on (17), the result of Theorem 2 is

$$\nu^{k,l}\left(\mathbf{X}^k\right) \approx \frac{1}{N^t} \prod_{j=1}^{t} \sum_{i=1}^{N} \delta\left(\mathbf{x}_j^k - \mathbf{x}_{j,i}^{k,l}\right). \quad (18)$$

Theorem 3 provides $\alpha^{l+1}(\cdot)$ based on (18). Given the PF approximation to $\nu^{k,l}(\cdot)$, we cannot evaluate $\nu^{k,l}(\cdot)$ directly because the PDF is represented by Dirac delta functions. There are several solutions to this. One is to regularise the PF approximation (18) [5]. In this case, (18) becomes the product of a Gaussian mixtures with N components. In order to lower the computational burden, we can merge some of these components [12]. However, in this paper, we take a simpler approach that consists of directly approximating (18) by independent Gaussian PDFs by moment matching.

We have

$$\nu^{k,l}\left(\mathbf{X}^k\right) = \nu_1^{k,l}\left(\mathbf{x}_1^k\right) ... \nu_t^{k,l}\left(\mathbf{x}_t^k\right) \quad (19)$$

$$\nu_j^{k,l}\left(\mathbf{x}_j^k\right) \approx \mathcal{N}\left(\mathbf{x}_j^k; \overline{\mathbf{x}}_j^{k,l}, \mathbf{P}_j^{k,l}\right) \quad (20)$$

where

$$\overline{\mathbf{x}}_j^{k,l} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{j,i}^{k,l} \quad (21)$$

$$\mathbf{P}_j^{k,l} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_{j,i}^{k,l} - \overline{\mathbf{x}}_j^{k,l}\right)\left(\mathbf{x}_{j,i}^{k,l} - \overline{\mathbf{x}}_j^{k,l}\right)^T. \quad (22)$$

Therefore, using Theorem 3, we get

$$\alpha^{l+1}\left(\mathbf{X}^k\right) \approx \frac{\prod_{j=1}^{t} \mathcal{N}\left(\mathbf{x}_j^k; \overline{\mathbf{x}}_j^{k,l}, \mathbf{P}_j^{k,l}\right)}{\sum_{p=1}^{t!} \prod_{j=1}^{t} \mathcal{N}\left(\mathbf{x}_{\phi_{p,j}}^k; \overline{\mathbf{x}}_j^{k,l}, \mathbf{P}_j^{k,l}\right)}. \quad (23)$$

2) MCMC algorithm to sample from $\pi^{k,l+1}(\cdot)$: In the previous subsection, we indicated how to approximate $\alpha^{l+1}(\cdot)$ based on the PF approximation to $\pi^{k,l}(\cdot)$. In this section, we design an MCMC to sample from $\pi^{k,l+1}(\cdot)$, which is given by (16). In principle, as we know the target distribution, we can use any MCMC algorithm to obtain samples, e.g., Metropolis-Hastings. Nevertheless, we can perform the MCMC sampling much more efficiently because of the specific characteristics of the target PDF (16).

In any MCMC algorithm, we need to evaluate the target PDF up to a proportionality constant. When we evaluate (16) for a state $\left(\mathbf{X}^k, \mathbf{a}\right)$, we have to compute the same terms we would calculate to evaluate (16) for any its permutations $\left(\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right), \mathbf{a}\right)$ $p = 1, ..., t!$. Therefore, as evaluating all the permutations comes at practically no cost, it is of high interest to develop an MCMC algorithm that accounts for all the permutations at the same step. The additional benefit is that this is a way to select the best labelling of each particle according to the PDF.

The MCMC algorithm developed in this paper does not provide any moves in the auxiliary variable $\mathbf{a}$. Conditioned on $\mathbf{a}$, we perform the moves in the state. In order to design this MCMC algorithm, we need to design a transition rule with respect to which the target PDF is invariant [13]. For a given $\mathbf{X}^k$, we propose the following transition rule to the next state $\mathbf{Y}^k$. First, we sample $\tilde{\mathbf{X}}^k$ from a density $q\left(\cdot \left| \mathbf{X}^k \right.\right)$. Then, we set $\mathbf{Y}^k = \Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right)$ or $\mathbf{Y}^k = \Gamma_{\boldsymbol{\phi}_p}\left(\tilde{\mathbf{X}}^k\right)$ with probabilities β_p' and $\tilde{\beta}_p'$ for $p = 1, ..., t!$

$$\mathbf{Y}^k = \begin{cases} \Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right) & \beta_p' = \frac{\beta_p}{\sum_{p=1}^{t!} \beta_p + \sum_{p=1}^{t!} \tilde{\beta}_p} \\ \Gamma_{\boldsymbol{\phi}_p}\left(\tilde{\mathbf{X}}^k\right) & \tilde{\beta}_p' = \frac{\tilde{\beta}_p}{\sum_{p=1}^{t!} \beta_p + \sum_{p=1}^{t!} \tilde{\beta}_p} \end{cases} \quad (24)$$

where

$$\beta_p = \pi^{k,l+1}\left(\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right), \mathbf{a}\right) \quad (25)$$

$$\tilde{\beta}_p = \pi^{k,l+1}\left(\Gamma_{\boldsymbol{\phi}_p}\left(\tilde{\mathbf{X}}^k\right), \mathbf{a}\right) \quad (26)$$

and $\pi^{k,l+1}(\cdot)$ is given by (16). It can be shown that this transition rule is invariant if $q(\cdot|\cdot)$ is symmetric and therefore

Table II: Steps of IT-MCMC

- We start with an evenly distributed PF approximation to $\pi^k(\cdot)$, which is given by $\left(\mathbf{X}_i^k, \mathbf{a}_i\right)\ i = 1, ..., N$.
- Set $\mathbf{X}_i^{k,0} = \mathbf{X}_i^k\ i = 1, ..., N$.
- For $l = 0, ..., L-1$
  - Calculate $\overline{\mathbf{x}}_j^{k,l}$, $\mathbf{P}_j^{k,l}\ j = 1, ..., t$ using (21) and (22).
  - Calculate $\alpha^{l+1}(\cdot)$ using (23).
  - Set $\mathbf{X}_i^{k,l+1} = \mathbf{X}_i^{k,l}\quad i = 1, ..., N$
  - For $h = 0, ..., H-1$
    - For each particle $i = 1, ..., N$
      - Sample $\tilde{\mathbf{X}}_i^{k,l+1}$ from $q\left(\cdot \left| \mathbf{X}_i^{k,l+1}\right.\right)$.
      - Calculate β_p for $p = 1, ..., t!$ using (25) by substituting $\mathbf{X}_i^{k,l+1}$ and $\mathbf{a}_i$ into and (16).
      - Calculate $\tilde{\beta}_p$ for $p = 1, ..., t!$ using (26) by substituting $\tilde{\mathbf{X}}_i^{k,l+1}$ and $\mathbf{a}_i$ into and (16).
      - Select $\mathbf{Y}_i^{k,l+1}$ among $\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}_i^{k,l+1}\right)$ and $\Gamma_{\boldsymbol{\phi}_p}\left(\tilde{\mathbf{X}}_i^{k,l+1}\right)$ $p = 1, ..., t!$ with probabilities given by (24).
      - Set $\mathbf{X}_i^{k,l+1} = \mathbf{Y}_i^{k,l+1}$.

leads to a valid MCMC algorithm by using the detailed balance condition [13]. However, due to a lack of space, we cannot include this proof in this paper.

To sum up, the IT-MCMC has three design parameters: the number L of steps of the recursion explained in Section III, the number H of steps that allow for the burn-in period of the MCMC algorithm to get samples from $\pi^{k,l}(\cdot)$ and the transition density $q\left(\cdot \left| \mathbf{X}^k\right.\right)$. Finally, the steps of the IT-MCMC algorithm are given in Table II.

V. Numerical simulations

In this section, we evaluate the performance of IT-MCMC against usual MCMC (U-MCMC) steps, which do not use the proposed iteration of Section III. We use the PP method to sample from (15) as it has been shown to outperform other MTT PF filters [3] and the IJOID method [6] cannot be applied to our scenario. The U-MCMC steps correspond to the Metropolis-Hastings algorithm [5].

We consider three targets that are being tracked by a sensor network. The state vector of the jth target at time k is $\mathbf{x}_j^k = \left[p_{x,j}^k, \dot{p}_{x,j}^k, p_{y,j}^k, \dot{p}_{y,j}^k\right]^T$. The dynamic model of the target is the nearly-constant velocity model:

$$\mathbf{x}_j^{k+1} = \mathbf{F}\mathbf{x}_j^k + \mathbf{v}^k$$
$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}$$

where $\otimes$ is the Kronecker product, $\mathbf{I}_n$ is the identity matrix of size n and $\mathbf{v}^k$ is the process noise at time k. The process noise is zero-mean Gaussian distributed with covariance matrix

$$\mathbf{Q} = q\mathbf{I}_2 \otimes \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix} \tag{27}$$

where q is a parameter of the model.

The target trajectories are shown in Figure 4. There are $M = 252$ sensors, so the measurement vector at time k is $\mathbf{z}^k = \left[z_1^k, ..., z_M^k\right]^T$ where z_j^k is the measurement of the jth sensor at time k. Sensor m is located at $\left[\xi_{x,m}, \xi_{y,m}\right]^T$ and measures an

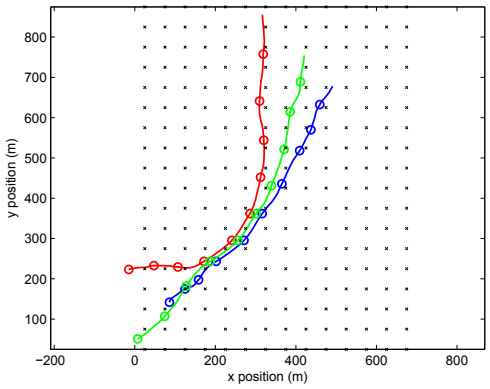


Figure 4: Scenario of the simulations. The target positions every ten time steps is represented by a circumference. The targets move from left to right. The sensor positions are represented by black crosses.

Table III: Parameters of the simulation

Parameter	Value
τ	0.5 s
q	3.24 $\mathrm{m}^2/\mathrm{s}^3$
σ_s^2	1
P_0	6.31 W (8 dBW)
d_0	20 m

acoustic signal emitted by the target with measurement model

$$z_m^k = \sqrt{\sum_{j=1}^{t} h_m\left(\mathbf{x}_j^k\right) + w_m^k}$$

where

$$h_m\left(\mathbf{x}_j^k\right) = \begin{cases} \frac{P_0 d_0^2}{d_m^2\left(\mathbf{x}_j^k\right)} & d_m^2\left(\mathbf{x}_j^k\right) > d_0^2 \\ P_0 & d_m^2\left(\mathbf{x}_j^k\right) \leq d_0^2 \end{cases}$$

and w_m^k is an independent zero-mean Gaussian noise with variance σ_s^2, P_0 is the saturation power, d_0 is the distance at which this saturation power is produced and $d_m^2\left(\mathbf{x}_j^k\right)$ is the square distance from the target $\mathbf{x}_j^k$ to the mth sensor

$$d_m^2\left(\mathbf{x}_j^k\right) = \left(p_{x,j}^k - \xi_{x,m}\right)^2 + \left(p_{y,j}^k - \xi_{y,m}\right)^2. \tag{28}$$

Both the U-MCMC and IT-MCMC use the transition density

$$q\left(\tilde{\mathbf{X}}^k \left| \mathbf{X}^k\right.\right) = \mathcal{N}\left(\tilde{\mathbf{X}}^k; \mathbf{X}^k, \mathbf{I}_t \otimes \mathbf{Q}\right). \tag{29}$$

To estimate the target states, we approximate the minimum mean square optimal subpattern assignment (MSOSPA) estimator [10], which uses the OSPA metric without cut-off distance, Euclidean distance and $p = 2$ [11]. The minimum MSOSPA estimator corresponds to the mean of the RFS posterior in a region of a Voronoi diagram. Therefore, we can use k-means clustering on the particles as in [14]. Importantly, this clustering is only performed to estimate target states, it does not affect the approximation of the posterior PDF at the following time steps as done in [14].

The algorithms performances are evaluated based on Monte Carlo simulation with 300 runs. The simulation parameters those given in Table III. The prior PDF of the jth target at time step 0 is $\mathcal{N}\left(\mathbf{x}_j^0; \overline{\mathbf{x}}_j^0, 5\mathbf{I}_4\right)$ where $\mathbf{I}_{n_x}$ is the identity matrix

Table IV: IT-MCMC parameters

Algorithm	L	H
IT-MCMC 1	1	20
IT-MCMC 2	2	10
IT-MCMC 3	3	6
IT-MCMC 4	4	5

of size $n_x \times n_x$ and $\overline{\mathbf{x}}_j^0$ is the expected position of the jth target at time 0. In each Monte Carlo run, $\overline{\mathbf{x}}_j^0$ is drawn from Gaussian distribution whose mean is the true target position and covariance matrix $5\mathbf{I}_4$.

We evaluate the performance of the algorithms with 300 and 500 particles. The U-MCMC algorithm uses 20 steps. As shown in Table IV, we have implemented 4 versions of IT-MCMC that differ in their parameters L and H, see Table II. All of them have around 20 MCMC steps such that the number of MCMC steps remains unaltered.

The root MSOSPA (RMSOSPA) position error of the algorithms are shown in Figure 5. IT-MCMC algorithms outperform U-MCMC. The improvement mainly happens after the targets have been in close proximity. In addition, the higher L is, slightly better performance is achieved although it is enough to make $L = 1$ to have a big increase in performance. While the performances for 300 and 500 particles are practically the same, we want to remark that the execution times for IT-MCMC algorithms with 300 particles are lower (from 7.9 to 8.64 s) than for U-MCMC with 500 particles (9.35 s).

VI. Conclusions

In this paper, we have used the fact that we can choose any labelled density within the RFS posterior family to develop an algorithm that improves the posterior independence approximation. This is of high interest in particle filtering due to the higher performance of PFs with PIA for a reasonable number of particles. The search of the labelled density with more independent target states than the original one is performed using MCMC steps. The benefits of this algorithm are expected to happen when targets get in close proximity and separate.

An interesting line of future work is to extend this work to an unknown and variable number of targets. In this case, the objective is to obtain the closest labelled multi-Bernoulli density to any labelled RFS density that belongs to the unlabelled posterior RFS family.

VII. Acknowledgements

This work was supported by the Australian Research Council under Discovery Project DP130104404.

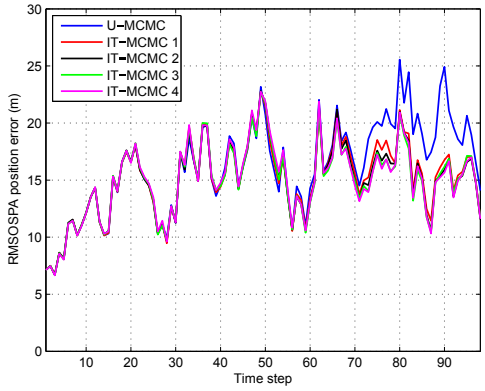
Appendix A

In this appendix, we prove (7). First, we prove $\Leftarrow$

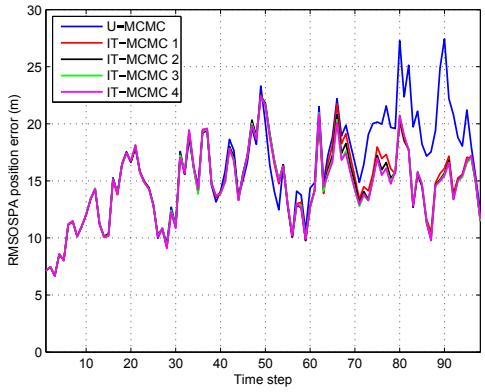
$$\pi^k\left(\mathbf{X}^k\right) = \alpha\left(\mathbf{X}^k\right)\breve{\pi}^k\left(\left\{\mathbf{x}_1^k, ..., \mathbf{x}_t^k\right\}\right)$$

$$\sum_{i=1}^{t!}\pi^k\left(\Gamma_{\boldsymbol{\phi}_i}\left(\mathbf{X}^k\right)\right) = \breve{\pi}^k\left(\left\{\mathbf{x}_1^k, ..., \mathbf{x}_t^k\right\}\right)\sum_{i=1}^{t!}\alpha\left(\Gamma_{\boldsymbol{\phi}_i}\left(\mathbf{X}^k\right)\right)$$

$$\sum_{i=1}^{t!}\pi^k\left(\Gamma_{\boldsymbol{\phi}_i}\left(\mathbf{X}^k\right)\right) = \breve{\pi}^k\left(\left\{\mathbf{x}_1^k, ..., \mathbf{x}_t^k\right\}\right)$$



(a)



(b)

Figure 5: RMSOSPA position error against time using the PP method (a) $N = 300$ (b) $N = 500$. IT-MCMC algorithms outperform U-MCMC.

so $\pi^k(\cdot) \in \mathcal{R}_{\breve{\pi}^k}$. Now, we prove $\Rightarrow$. If $\pi^k(\cdot) \in \mathcal{R}_{\breve{\pi}^k}$, we choose

$$\alpha\left(\mathbf{X}^k\right) = \frac{\pi^k\left(\mathbf{X}^k\right)}{\breve{\pi}^k\left(\left\{\mathbf{x}_1^k, ..., \mathbf{x}_t^k\right\}\right)}$$

$$\sum_{i=1}^{t!}\alpha\left(\Gamma_{\boldsymbol{\phi}_i}\left(\mathbf{X}^k\right)\right) = \frac{\sum_{i=1}^{t!}\pi^k\left(\Gamma_{\boldsymbol{\phi}_i}\left(\mathbf{X}^k\right)\right)}{\breve{\pi}^k\left(\left\{\mathbf{x}_1, ..., \mathbf{x}_t\right\}\right)} = 1.$$

Appendix B

In this appendix, we prove Theorem 2. We want to find $\nu_1^{k,\star}(\cdot), ..., \nu_t^{k,\star}(\cdot)$ such that

$$\nu_1^{k,\star}(\cdot), ..., \nu_t^{k,\star}(\cdot) = \underset{\nu_1^k(\cdot),...,\nu_t^k(\cdot)}{\arg\min}\ \mathrm{D}\left(\pi^k(\cdot)\left\|\nu_1^k(\cdot), ..., \nu_t^k(\cdot)\right.\right)$$

$$= \underset{\nu_1^k(\cdot),...,\nu_t^k(\cdot)}{\arg\min}\int \pi^k\left(\mathbf{X}^k\right)\log\frac{\pi^k\left(\mathbf{X}^k\right)}{\nu_1^k\left(\mathbf{x}_1^k\right), ..., \nu_t^k\left(\mathbf{x}_t^k\right)}d\mathbf{X}^k.$$

We can write

$$\nu_i^{k,\star}(\cdot) = \underset{\nu_i^k(\cdot)}{\arg\min} -\int \pi_i^k\left(\mathbf{x}_i^k\right)\log\nu_i^k\left(\mathbf{x}_i^k\right)d\mathbf{x}_i^k$$

$$= \underset{\nu_i^k(\cdot)}{\arg\min}\int \pi_i^k\left(\mathbf{x}_i^k\right)\log\frac{\pi_i^k\left(\mathbf{x}_i^k\right)}{\nu_i^k\left(\mathbf{x}_i^k\right)}d\mathbf{x}_i^k \qquad (30)$$

where $\pi_i^k(\cdot)$ is the marginal PDF of $\mathbf{x}_i^k$ in $\pi^k(\cdot)$. By the definition of entropy, $\nu_i^{k,\star}(\cdot) = \pi_i^k(\cdot)$.

APPENDIX C

In this appendix, we prove Theorem 3. In the following, we make use of the calculus of variations to minimise the functional [15]

$$L[\alpha] = \mathrm{D}\left(\alpha(\cdot)\cdot\breve{\pi}^k(\{\cdot\})\,\middle\|\,\nu^k(\cdot)\right) \tag{31}$$

subject to the constraint (8). A similar proof for two targets is given in [16].

We can find a stationary point of $L[\cdot]$ by setting [15]

$$\left.\frac{dL[\alpha+\epsilon\eta]}{d\epsilon}\right|_{\epsilon=0} = 0$$

where ϵ is a constant and $\eta(\cdot)$ is a function such that the function $\alpha(\cdot)+\epsilon\eta(\cdot)$ also meets the constraint (8), which implies that

$$\sum_{p=1}^{t!}\eta\left(\Gamma_{\boldsymbol{\phi}_p}\left(\mathbf{X}^k\right)\right) = 0. \tag{32}$$

In this derivation, we omit the time index of the state. We get that

$$\begin{aligned}\frac{dL[\alpha+\epsilon\eta]}{d\epsilon} = \int\Big[&\eta(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\\ &\log\frac{(\alpha(\mathbf{X})+\epsilon\eta(\mathbf{X}))\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})}{\nu^k(\mathbf{X})}\\ &+\eta(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\Big]\,d\mathbf{X}.\end{aligned} \tag{33}$$

Let us calculate

$$\begin{aligned}&\int\eta(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\,d\mathbf{X}\\ &\quad= \frac{1}{t!}\int\sum_{p=1}^{t!}\eta\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\,d\mathbf{X} \qquad (34)\\ &\quad= 0 \qquad (35)\end{aligned}$$

where we have used (32). Then,

$$\begin{aligned}\left.\frac{dL[\alpha+\epsilon\eta]}{d\epsilon}\right|_{\epsilon=0} = &\int\eta(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\\ &\log\frac{\alpha(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})}{\nu^k(\mathbf{X})}d\mathbf{X}.\end{aligned}$$

Expanding the logarithm and using (35) again, we get

$$\left.\frac{dL[\alpha+\epsilon\eta]}{d\epsilon}\right|_{\epsilon=0} = \int\eta(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\log\frac{\alpha(\mathbf{X})}{\nu^k(\mathbf{X})}d\mathbf{X}.$$

We want to find $\alpha(\cdot)$ that meets the constraint (8) for any $\eta(\cdot)$ with the constraint (32) such that

$$\int\eta(\mathbf{X})\,\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\log\frac{\alpha(\mathbf{X})}{\nu^k(\mathbf{X})}d\mathbf{X} = 0.$$

We can also write

$$\int\sum_{p=1}^{t!}\eta\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)\breve{\pi}^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})\log\frac{\alpha\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)}{\nu^k\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)}d\mathbf{X} = 0.$$

Using the same argument as in (35), the equation is zero if $\alpha\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)/\nu^k\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)$ does not depend on p, i.e., it has the same value regardless of the permutation. This is achieved if $\alpha(\mathbf{X}) = a(\{\mathbf{x}_1,...,\mathbf{x}_t\})\,\nu^k(\mathbf{X})$ where $a(\cdot)$ does not depend on p. We also know that $\alpha(\cdot)$ must meet the constraint (8). As a result, the stationary point is

$$\alpha^\star(\mathbf{X}) = \frac{\nu^k(\mathbf{X})}{\sum_{p=1}^{t!}\nu^k\left(\Gamma_{\boldsymbol{\phi}_p}(\mathbf{X})\right)}. \tag{36}$$

We have to compute the second derivative to show that it is a minimum. We get

$$\left.\frac{d^2L[\alpha+\epsilon\eta]}{d\epsilon^2}\right|_{\epsilon=0} = \int\eta^2(\mathbf{X})\,\frac{\pi^k(\{\mathbf{x}_1,...,\mathbf{x}_t\})}{\alpha^\star(\mathbf{X})}d\mathbf{X}.$$

Assuming that the integrand is absolute integrable, this quantity is positive as the integrand is always positive. Therefore, this a a minimum. The problem is convex so this is the global minimum as pointed out [16]. Substituting (36) into (7) completes the proof.

REFERENCES

[1] 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.

[2] B. T. Vo and B. N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, July 2013.

[3] A. F. García-Fernández, J. Grajal, and M. R. Morelande, "Two-layer particle filter for multiple target detection and tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1569–1588, July 2013.

[4] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA filter for multitarget tracking," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4677–4691, Oct. 2011.

[5] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[6] W. Yi, M. R. Morelande, L. Kong, and J. Yang, "A computationally efficient particle filter for multitarget tracking using an independence approximation," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 843–856, 2013.

[7] 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.

[8] 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/

[9] J. E. Moyal, "The general theory of stochastic population processes," *Acta Mathematica*, vol. 108, pp. 1–31, 1962.

[10] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *13th Conference on Information Fusion*, July 2010, pp. 1–8.

[11] 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.

[12] D. Salmond, "Mixture reduction algorithms for point and extended object tracking in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, pp. 667–686, April 2009.

[13] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[14] 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.

[15] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*. Prentice-Hall, 1963, translated by R. A. Silverman.

[16] L. Svensson, D. Svensson, and M. Guerriero, "Set JPDA filter for multiple target tracking," Chalmers University of Technology, Tech. Rep. R012/2010, 2010.