# Rao-Blackwellised Particle Filtering in Random Set Multitarget Tracking

Matti Vihola
Datactica Ltd, Hermiankatu 1, FI-33720 Tampere, Finland
E-mail: forename.surname "at" iki.fi

***Abstract*— This article introduces a Rao-Blackwellised particle filtering (RBPF) approach in the finite set statistics (FISST) multitarget tracking framework. The RBPF approach is proposed in such a case, where each sensor is assumed to produce a sequence of detection reports each containing either one single-target measurement, or a "no detection" report. The tests cover two different measurement models: a linear-Gaussian measurement model, and a nonlinear model linearised in the extended Kalman filter (EKF) scheme. In the tests, Rao-Blackwellisation resulted in a significant reduction of the errors of the FISST estimators when compared to a previously proposed direct particle implementation. In addition, the RBPF approach was shown to be applicable in nonlinear bearings-only multitarget tracking.**

***Index Terms*— Finite set statistics, multitarget tracking, Rao-Blackwellised particle filter**

## I. Introduction

THE finite set statistics (FISST) approach to multitarget tracking was introduced to the tracking community almost a decade ago [1]. While theoretically solid, the FISST approach has been abandoned by many tracking researchers and engineers due to the intractable computations related to it. Lately, the FISST formulation has gained much attention, due to the recently developed computational approximation strategies: sequential Monte Carlo (SMC) sampling algorithms [2]–[7] and the probability hypothesis density (PHD) approximation [8]–[10]. The SMC algorithms that have been proposed for FISST tracking have been quite simple and inefficient so far. This article proposes a more efficient SMC approach for FISST filtering: the Rao-Blackwellised particle filter (RBPF).

Rao-Blackwellisation can be employed if the model is of a specific form. In the case of certain type of a restricted model family, one can take advantage on the underlying structure of the model in order to derive a more efficient SMC implementation. In this article, the conditionally linear-Gaussian model is considered. The linear-Gaussian restriction can be relaxed to some extent with known approximation methods, such as the extended Kalman filter (EKF) [11] or the unscented Kalman filter (UKF) [12].

While the PHD approximation approach for FISST filtering [9] has drawn great attention lately, this article does not involve this specific approximation. Even though the PHD approximation is tempting due to some of its computational properties[1], it is still a crude approximation. For example, it discards any inter-target dependencies. The method introduced in this article may be useful in some cases where the PHD approach fails to produce good results[2]. In addition, the PHD approach has been proposed for quite a different sensor model than the one considered in this article.

### A. Goal

The main goal of multitarget tracking is, in general, to produce an estimate of a multitarget state based on a sequence of sensor detection reports. The multitarget state consists of a number of targets in the surveillance region having states (e.g. positions, velocities, ...). Due to limited data rate and unknown movement, appearance, and disappearance behaviour of the targets, the multitarget state cannot be inferred exactly, but remains uncertain. For example, in passive surveillance, where the data consists of bearings-only measurements, this uncertainty is often inherently large due to low data rate and/or poor data quality. The algorithms proposed in this article are targeted to the application areas similar in nature.

The specific application area that is addressed in this article covers sensors producing detection reports with one single-target measurement. In addition, sensors can produce "no detection" reports. There can be multiple sensors whose detection reports are processed sequentially[3]. The reporting intervals of the sensors are assumed completely uncharacterised, and allowed to be irregular. This kind of behaviour modelling is motivated by independently operating sensors and a network-centric environment, where the transmission links can be relatively unreliable.

### B. Contribution and Related Work

The main contribution of this article is the introduction of Rao-Blackwellised particle filtering in the context of FISST multitarget tracking. The multitarget measurement

[1]The PHD filter prediction and correction equations do not depend on the number of targets.

[2]Mahler has stated that the PHD approximation inherently presumes relatively high SNR sensor [9].

[3]The measurements are assumed to arrive in chronological order—the out-of-sequence issues are not addressed in this article.

model that is employed is new, at least in the FISST context. However, similar models has been described in [4], [5], [13]–[15]. The multitarget model that is used in the article is a common one including independent target motions, births[4] and deaths (see, e.g., [4], [5]), with the exception of a new "symmetrisation" introduced in order to minimise the bias in the estimation of the number of targets.

The Rao-Blackwellised multitarget tracking algorithms derived in this article have much in common with the algorithm proposed in [15] in the case of known number of targets, and extended to unknown number of targets in [14]. In fact, the algorithm described by [14] can be considered a special case of the algorithms proposed in this article—although they derived and presented their approach without the notion of random sets, and having different multitarget measurement and dynamic models. The sampling scheme of [14] resembles most the importance distribution "RBPF C" described in Sect. IV-C.

Due to the long history of multitarget tracking, there have been numerous articles describing different methods for tackling the problem (see, e.g., [9] for a review). From the traditional methods, Multiple Hypothesis Tracking (MHT) as proposed by [13], [16], resembles most the FISST RBPF proposed in this article. MHT differs in the sense that it has a deterministic (and typically heuristic) hypothesis-pruning algorithms, whereas the RBPF algorithms have random "hypothesis-pruning", and are theoretically valid.[5]

Joint Probabilistic Data Association (JPDA) is another commonly-used multitarget tracking algorithm. JPDA was originally proposed only for tracking a known number of targets, but can be extended to the case of unknown number of targets by means of a track maintenance logic [17]. On the contrary to RBPF and MHT, JPDA restricts the multitarget posterior distribution to be unimodal (Gaussian). This restriction can be considered to be cumbersome for the purposes of such applications addressed by this article (see Sect. I-A).

### C. Notations

In this article, random variables are denoted with boldface, sets with capital letters, vectors are underlined, and matrices written with capital upright letter, such as $\boldsymbol{x}$, $C$, $\underline{m}$, and $\mathrm{A}$, respectively. These conventions are used together, so that, e.g., $\boldsymbol{X}$ is a random set. Sequences are often indexed using the shorthand notation $x_{a:b} = (x_a, x_{a+1}, \ldots, x_b)$. Cardinality of a set, i.e. the number of elements in the set, is denoted as $|C|$.

Probability measure induced by random variable $\boldsymbol{x}$ and the corresponding density are denoted as follows

$$P_{\boldsymbol{x}}(C) \triangleq P(\boldsymbol{x} \in C) = \int_C p_{\boldsymbol{x}}(x)\mu(\mathrm{d}x)$$

where $P$ is the common probability measure, $P_{\boldsymbol{x}}$ is the induced probability measure, and $p_{\boldsymbol{x}}$ is the probability density function of $\boldsymbol{x}$ against a measure[6] $\mu$, which is clear from the context. The conditional probability measures and densities are denoted similarly,

$$P_{\boldsymbol{x}|\boldsymbol{y}}(C \mid y) \triangleq P(\boldsymbol{x} \in C \mid \boldsymbol{y} = y) = \int_C p_{\boldsymbol{x}|\boldsymbol{y}}(x \mid y)\mu(\mathrm{d}x)$$

Measures are sometimes written without argument; for example one should read a statement like $P_{\boldsymbol{x}} = P_{\boldsymbol{y}}$, so that $P_{\boldsymbol{x}}(C) = P_{\boldsymbol{y}}(C)$ for all (measurable) sets $C$.

The Dirac measure[7] concentrated at $x$ is denoted as $\delta_x$, and $U(C; a, b)$, $N(C; \underline{m}, \mathrm{P})$ refer to measures corresponding to, respectively, a uniform distribution in the interval $[a, b]$, and a Gaussian distribution with the mean vector $\underline{m}$ and the covariance matrix $\mathrm{P}$. Discrete distributions are denoted using their probability mass functions, i.e. $q_j(\cdot)$ refers to a discrete distribution such that $P(\cdot = j) = q_j$. The notation "$\sim$" is used to denote how a random variable is distributed, possibly including a dependency on another random variable; e.g. $\boldsymbol{z} \sim N(\cdot; h(\boldsymbol{y}), \mathrm{P})$ should be read so that the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{y} = y$ is Gaussian with mean $h(y)$ and covariance $\mathrm{P}$.

### D. Structure

The rest of this article is organised as follows. First, Sect. II introduces Rao-Blackwellised particle filtering in general state-spaces before the proposed FISST model is described in Sect. III. After that, Sect. IV shows how the general Rao-Blackwellisation of Sect. II can be applied to the FISST model in Sect. III, and describes three algorithms implementing different importance distributions. Visualisation issues and the extracted estimates are discussed in Sect. V, and Sect. VI covers the experimental setup and the obtained results. Finally, in Sect. VII conclusions are drawn and some ideas for future development are discussed. The appendix contains derivation of the PHD in the case of a Rao-Blackwellised random set particle filter.

## II. Rao-Blackwellised Particle Filtering

In general, Rao-Blackwellised particle filtering refers to a sequential Monte Carlo method, in which only some variables are sampled, while the other are handled analytically [18]. Rao-Blackwellisation can result in a tremendous decrease in the variance of a Monte Carlo estimate when compared to a direct particle implementation without Rao-Blackwellisation. On the other hand,

[4] In this article, spawning of new targets from the existing ones is not considered.

[5] That is, as the number of samples increases, the posterior expectation estimate converges (with minimal assumptions) to the true posterior expectation value. On the contrary to the philosophy behind many MHT implementations, RBPF is not meant to find a "winning hypothesis", but to estimate multitarget posterior expectations.

[6] Such a density function is assumed to exist without explicit notification, as well as all the conditional probability measures and densities. In the case that $\mu$ is the Lebesgue measure, the differential "$\mu(\mathrm{d}x)$" is written as usual "$\mathrm{d}x$".

[7] Dirac measure is defined such that $\delta_x(C) = 1$ for any set $C$ containing $x$, and $\delta_x(C) = 0$ if $x \notin C$.

Rao-Blackwellisation restricts the form of the estimation model.

Perhaps the most common model with which RBPF has been applied is the conditional linear-Gaussian (CLG) model. In a CLG model, given the values of certain random variables, the model reduces into a linear-Gaussian one, admitting exact inference within the well-known Kalman filtering framework. The idea of RBPF was presented in the special case of a CLG model by Chen and Liu [19]. They called the method the *mixture Kalman filter*.

Suppose that the state process is a Markov chain $(\boldsymbol{z}_k)_{k=0}^{\infty}$, and the sequence of measurements $(\boldsymbol{y}_k)_{k=1}^{\infty}$ are independent conditional on the state process,

$$P_{\boldsymbol{y}_k|\boldsymbol{z}_{a:b},\boldsymbol{y}_{c:d\setminus k}} = P_{\boldsymbol{y}_k|\boldsymbol{z}_k}$$

for all integers $k \geq 1$, $0 \leq a \leq k \leq b$ and $1 \leq c \leq d$.[8] In addition, suppose that $\boldsymbol{z}_k$ can be decomposed so that $\boldsymbol{z}_k = (\boldsymbol{r}_k, \boldsymbol{x}_k)$, and the conditional expectations $\mathsf{E}\left[h(\boldsymbol{z}_{0:k}) \mid \boldsymbol{y}_{1:k} = y_{1:k}, \boldsymbol{r}_{0:k} = r_{0:k}\right]$ can be computed analytically for the functions of interest $h$. Furthermore, suppose that $P_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}$ can be approximated with $\widehat{P}_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}$ having $N$ weighted random samples,

$$\widehat{P}_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}(C \mid y_{1:k}) = \sum_{i=1}^{N} w^{(i)} \delta_{r_{0:k}^{(i)}}(C)$$

Now, since the posterior distribution can be approximated as follows

$$\begin{aligned} P_{\boldsymbol{r}_{0:k},\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}} &= P_{\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k},\boldsymbol{r}_{0:k}} P_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}} \\ &\approx P_{\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k},\boldsymbol{r}_{0:k}} \widehat{P}_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}} \end{aligned}$$

one can form an estimate $I_N(h)$ approximating $\mathsf{E}\left[h(\boldsymbol{z}_{0:k}) \mid \boldsymbol{y}_{1:k} = y_{1:k}\right]$ as follows [18]

$$I_N(h) \triangleq \sum_{i=1}^{N} w^{(i)} \mathsf{E}\left[h(\boldsymbol{z}_{0:k}) \,\middle|\, \boldsymbol{y}_{1:k} = y_{1:k}, \boldsymbol{r}_{0:k} = r_{0:k}^{(i)}\right] \quad (1)$$

In particular, the approximation $\widehat{P}_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}$ can be formed so that $r_{0:k}^{(i)}$ are independent samples from an importance distribution $Q_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}$ and the weights $w^{(i)}$ are computed as follows

$$w^{(i)} = \frac{\widetilde{w}^{(i)}}{\sum_{i=1}^{N} \widetilde{w}^{(i)}}, \qquad \text{where} \qquad \widetilde{w}^{(i)} \propto \frac{p(r_{0:k}^{(i)} \mid y_{1:k})}{q(r_{0:k}^{(i)} \mid y_{1:k})}$$

where $p$ and $q$ are the densities of the measures $P_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}$ and $Q_{\boldsymbol{r}_{0:k}|\boldsymbol{y}_{1:k}}$ with respect to a common measure (see Sect. I-C). In such a case, the estimate $I_N(h)$ converges to $\mathsf{E}\left[h(\boldsymbol{z}_{0:k}) \mid \boldsymbol{y}_{1:k} = y_{1:k}\right]$ as the number of samples $N$ is increased [18].

In particular, if the importance distribution $q$ is assumed to be decomposable as follows

$$q(r_{0:k} \mid y_{1:k}) = q_0(r_0) \prod_{j=1}^{k} q_j(r_j \mid y_{1:j}, r_{0:j-1})$$

[8] Backslash is a "set difference", i.e. the notation $\boldsymbol{y}_{c:d\setminus k}$ stands for "all $\boldsymbol{y}_i$ where $i$ is between $c$ and $d$, excluding $k$".

where $q_0$ and $q_k$ are (conditional) probability densities. Then, the random samples $r_{0:k}$ can be sampled so that $r_0^{(i)} \sim q_0$ and the rest are generated recursively as follows

$$r_k^{(i)} \sim q_k(\cdot \mid r_{0:k-1}^{(i)}, y_{1:k})$$

In this case, the importance weights can be computed sequentially as well, as follows [20, p. 501]

$$\begin{aligned} \widetilde{w}_k^{(i)} \propto \widetilde{w}_{k-1}^{(i)} & p_{\boldsymbol{y}_k|\boldsymbol{r}_{0:k},\boldsymbol{y}_{1:k-1}}(y_k \mid r_{0:k}^{(i)}, y_{1:k-1}) \\ & \frac{p_{\boldsymbol{r}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{y}_{1:k-1}}(r_k^{(i)} \mid r_{0:k-1}^{(i)}, y_{1:k-1})}{q_k(r_k^{(i)} \mid r_{1:k-1}^{(i)}, y_{1:k})} \end{aligned} \quad (2)$$

and obtaining the normalised weights $w_k^{(i)} = \widetilde{w}_k^{(i)} / \sum_j \widetilde{w}_k^{(j)}$.

The RBPF algorithm includes one more step common to all particle filters, the resampling or selection phase. Numerous different resampling strategies have been proposed in the literature [20]. In this article, the *adaptive resampling* strategy is used [21].

## III. The Random Set Model

The FISST multitarget tracking approach assumes a *multitarget model*, in which the statistical estimation (tracking) is performed. The multitarget model includes the estimated multitarget state as a random set $\boldsymbol{X}_k$. Similarly, the observations are assumed to arrive in detection reports having *a priori* unknown number of measurements. They are modelled as random sets $\boldsymbol{Y}_k$.

Practical model building and analysis of FISST models is done using the FISST calculus, derived in [1]. The most important underlying issue is the construction of the concepts of the *set derivative* and the *set integral*, which are special cases of general measure-theoretic integration and differentiation. In particular, they are defined as the integral and the Radon-Nikodym derivative with respect to the measure [4]

$$\mu(\mathcal{O}) = \delta_{\{\emptyset\}}(\mathcal{O}) + \sum_{k=1}^{\infty} \mu_k(\mathcal{O}) \quad \text{where} \quad \mu_k(\mathcal{O}) = \frac{\lambda^k(\overleftarrow{\eta}(\mathcal{O}_k))}{k!}$$

where $\mathcal{O}$ is a measurable collection of finite subsets of $\mathbb{R}^d$ $\mathcal{O}_k = \mathcal{O}_k \cap \mathcal{F}(k)$ with $\mathcal{F}(k)$ being the collection of all finite sets having exactly $k$ members, and $\lambda^k$ is the $k$-fold Lebesgue product measure. In addition, $\overleftarrow{\eta}$ is the preimage of the function $(s_1, \ldots, s_k) \mapsto \{s_1, \ldots, s_k\}$ mapping values from product space $[\mathbb{R}^d]^k$ to a finite subset of $\mathbb{R}^d$. That is, the set integral can be given as follows

$$\int f(X)\delta X \triangleq \int f \mathrm{d}\mu$$

The density $p_{\boldsymbol{X}}$ of the random set measure $P_{\boldsymbol{X}}$ is defined as described in Sect. I-C, so that

$$P_{\boldsymbol{X}}(\mathcal{C}) \triangleq P(\boldsymbol{X} \in \mathcal{C}) = \int_{\mathcal{C}} p_{\boldsymbol{X}}(X)\delta X = \int_{\mathcal{C}} p_{\boldsymbol{X}}(X)\mathrm{d}\mu(X)$$

where $\mathcal{C}$ is a (measurable) collection of finite sets, and $p_{\boldsymbol{X}} = \delta P_{\boldsymbol{X}}/\delta X = \mathrm{d}P_{\boldsymbol{X}}/\mathrm{d}\mu$ is the set derivative of $P_{\boldsymbol{X}}$ (i.e. a Radon-Nikodym derivative with respect to $\mu$).

An especially useful concept of FISST is the *belief-mass function*,

$$\beta_{\boldsymbol{X}}(C) \triangleq P(\boldsymbol{X} \subset C)$$

defined for all (measurable) $C \subset \mathbb{R}^d$. The belief-mass function determines the distribution $P_{\boldsymbol{X}}$ of the random set $\boldsymbol{X}$ uniquely. For detailed introduction to the concepts of FISST multitarget tracking, consult [1], and for the measure-theoretic formulation, see [4], [7].

The random set model proposed in this article consists of two parts: the multitarget dynamic model (a finite set valued Markov chain), and the multitarget measurement model (conditionally independent finite set measurements of the finite set state). Fig. 1 (a) shows the random set model as a Bayesian network[9]. Figs. 1 (b) and (c) show the details of the dynamic model and the measurement model, respectively, that are covered in detail in Sections III-A and III-B. The filtering with the model is discussed in Sect. III-C.

### A. The Multitarget Dynamic Model

The multitarget dynamic model is defined by the conditional probability measures $P_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}$. These conditional probabilities determine the state evolution process, which is the Markov chain $(\boldsymbol{X}_k)_{k=0}^{\infty}$. Without better prior information of target count or state, a rational choice of the prior distribution of $\boldsymbol{X}_0$ is to set $P(\boldsymbol{X}_0 = \emptyset) = 1$.

In the following subsections, the sub-models that build up $P_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}$ are described. First, Sect. III-A.1 describes a simple single-target dynamic model. Next, Sect. III-A.2 describes the birth model, i.e. the model how the targets appear in the surveillance region. Finally, Sect. III-A.3 describes the death model, i.e. the model how the targets disappear in the surveillance region. The described multitarget dynamic model is similar to what described in [4], [5], except that the birth and death models are built up so that bias in the estimated number of targets is avoided.

The single-target dynamic model in Sect. III-A.1 and the death model in Sect. III-A.3 determine the connection of $\boldsymbol{X}_{k-1}$, $\boldsymbol{d}_k$, and $\boldsymbol{S}_k$ in Fig. 1 (c). The birth model, described in Sect. III-A.2, characterises the part of the model with the variables $\boldsymbol{b}_k$ and $\boldsymbol{B}_k$. The new multitarget state $\boldsymbol{X}_k$ is formed from the set of the survived targets $\boldsymbol{S}_k$ and the set of the appeared new targets $\boldsymbol{B}_k$ by an obvious model

$$\boldsymbol{X}_k = \boldsymbol{S}_k \cup \boldsymbol{B}_k$$

which means that $\boldsymbol{B}_k$ and $\boldsymbol{S}_k$ are lumped together to form the new target state $\boldsymbol{X}_k$.

*1) Single-target dynamic model:* The single-target dynamic models for tracking purposes have been studied extensively [11], [23]. In this article, the single-target dynamic model is assumed to be linear-Gaussian[10] with the almost constant velocity model. The model can be given as follows [11].

$$P_{\underline{\boldsymbol{x}}_k|\underline{\boldsymbol{x}}_{k-1}}(C \mid \underline{x}_{k-1}) = N(C; \mathrm{A}_k \underline{x}_{k-1}, \mathrm{Q}_k) \tag{3}$$

The matrices $\mathrm{A}_k$ and $\mathrm{Q}_k$ can be given as follows.

$$\mathrm{A}_k = \begin{bmatrix} \mathrm{I} & \tau_k \mathrm{I} \\ 0 & \mathrm{I} \end{bmatrix}, \qquad \mathrm{Q}_k = \rho^2 \begin{bmatrix} (\tau_k^3/3)\mathrm{I} & (\tau_k^2/2)\mathrm{I} \\ (\tau_k^2/2)\mathrm{I} & \tau_k \mathrm{I} \end{bmatrix} \tag{4}$$

where $\tau_k = t_{k+1} - t_k$ is the time interval between $\underline{\boldsymbol{x}}_{k+1}$ and $\underline{\boldsymbol{x}}_k$, $\rho$ is the process noise parameter, I is a $d \times d$ identity matrix, and 0 is a zero matrix of the same size.

*2) Birth model:* The birth model of the targets is assumed to be as uninformative as possible (if no additional information is available). In addition, in this article such sensors are considered that produce detection reports on an irregular basis. The Poisson birth model is a good choice in this situation. The Poisson birth model can be given as follows

$$\beta_{\boldsymbol{B}_k}(C) = \sum_{j=0}^{\infty} b_j P_b(C)^j \tag{5}$$

where the terms $b_j = P(|\boldsymbol{B}_k| = j)$ are values of the Poisson distribution

$$b_j = \frac{(\eta \tau_k)^j}{j!} e^{-\eta \tau_k} \tag{6}$$

where $\eta$ is the birth rate parameter, and $\tau_k$ is the time difference of the $k-1$:th and $k$:th sensor report. The measure $P_b = N(\cdot; \underline{m}_0, \mathrm{P}_0)$ in Eq. (5) is the common initial distribution of all the born targets. Since the number of born targets follows a Poisson distribution determined by $b_j$, the expected number of born targets is $\eta\tau_k$ [24]. This implies that the use of the Poisson birth model allows the irregular update scheme, since the expected number of births is invariant of division to subintervals, $\eta(\tau_1 + \tau_2) = \eta\tau_1 + \eta\tau_2$.

*3) Death model:* In order to minimise the bias in the estimated number of targets, it is desired that the birth and the death models are "symmetric". In particular, it is required that whenever the number of targets is more than zero, the expected number of born and died targets is equal, or equivalently, the expected number of targets at time $k$ is equal to the number of targets at time $k-1$. That is,

$$\mathsf{E}\left[ |\boldsymbol{X}_k| - |\boldsymbol{X}_{k-1}| \;\middle|\; |\boldsymbol{X}_{k-1}| = m \right] = 0$$

which yields, assuming $|\boldsymbol{X}_k| = |\boldsymbol{B}_k \cup \boldsymbol{S}_k| \overset{\text{a.s.}}{=} |\boldsymbol{B}_k| + |\boldsymbol{S}_k|$ (which is satisfied when $\boldsymbol{B}_k \cap \boldsymbol{S}_k \overset{\text{a.s.}}{=} \emptyset$),

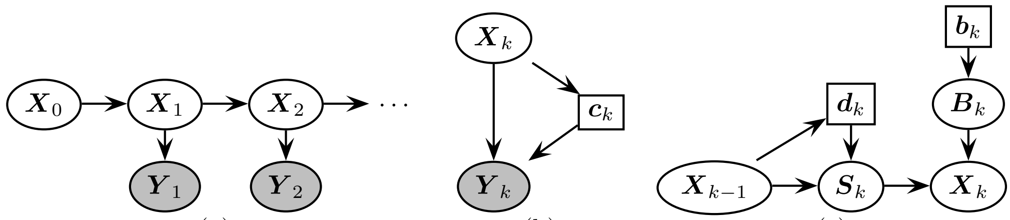
$$\mathsf{E}\left[ |\boldsymbol{X}_{k-1}| - |\boldsymbol{S}_k| \;\middle|\; |\boldsymbol{X}_{k-1}| = m \right] = \mathsf{E}\left[ |\boldsymbol{B}_k| \right] \tag{7}$$

whenever $m \geq 1$. In addition, it is assumed that the disappearance of different targets are independent of others. The conditional distribution of the number of non-surviving targets can be chosen to be the binomial distribution

$$P\left( |\boldsymbol{X}_{k-1}| - |\boldsymbol{S}_k| = n \;\middle|\; |\boldsymbol{X}_{k-1}| = m \right) = \binom{m}{n} (1-p_s)^n p_s^{m-n} \tag{8}$$

[9]For general information on Bayesian networks, see e.g. [22].

[10]This can be relaxed, e.g., by the use of EKF/UKF in the place of Kalman filter. This is discussed further below in Sect. VII.



(a) (b) (c)

Fig. 1: (a) Figure of a Bayesian network representing the random set model. (b) Details of the measurement model $P_{\boldsymbol{Y}_k|\boldsymbol{X}_k}$. (c) Details of the dynamic model $P_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}$. The nodes representing the observed variables are shaded. Rectangular nodes denote discrete random variables.

where $p_s$ is the probability of survival of a target, $n \leq m$, and $\binom{m}{n} \triangleq \frac{m!}{n!(m-n)!}$ is the binomial coefficient. The mean of the binomial distribution in Eq. (8) is [24]

$$\mathsf{E}\left[|\boldsymbol{X}_{k-1}| - |\boldsymbol{S}_k| \;\middle|\; |\boldsymbol{X}_{k-1}| = m\right] = m(1-p_s)$$

Thus, given the condition in Eq. (7), and remembering that the expected number of born targets in the Poisson birth model is $\eta\tau_k$, where $\tau_k$ is the time interval, the following condition for $p_s$ is obtained:

$$1 - p_s = \frac{\eta\tau_k}{m} \tag{9}$$

Note that if $m = 0$, the parameter $p_s$ is not needed, since there are no targets alive, and the only sensible choice is to set $P(\boldsymbol{S}_k = \emptyset \mid \boldsymbol{X}_{k-1} = \emptyset) = 1$. In addition, note that Eq. (9) can be satisfied for all $m \geq 1$ only if $\eta\tau_k \leq 1$. Since typical values of $\eta$ are small, $\eta\tau_k$ exceeds one only in rare cases, in which one can "reset" the system, e.g. by defining $p_s = 0$.

### *B. The Multitarget Measurement Model*

The measurement model that is proposed in this article differs from the usual scan-based model that is often used for radars. It is assumed that the tracking system obtains detection reports from independently operating sensors. Each report either contains one single-target measurement, e.g. one bearing measurement, or is a "no detection" report. The scanning patterns of the sensors are assumed random, not because real sensors are believed to operate randomly, but since the accurate scanning behaviour of the sensors is assumed to be unknown. This sensor model is motivated primarily on the behaviour of passive sensors that scan the surveillance space continuously, and report whenever a target is detected. It is expected that one cannot track a large number of targets using this model.[11] The model could be applied also to such sensors that include more than one measurement in a report, so that the measurements are processed sequentially, one after another.[12]

[11] The actual feasible number of targets depends, of course, on the specific parameters of the system. Using the parameter values in the example scenarios in Sect. VI, it is expected that the system cannot track more than, say, ten targets at a time.

[12] It must be noted, though, that this model may not be suitable, if the sensor produces a large amount of measurements per scan—such as radar typically does.

The measurement model is similar to the one derived in [4], [5]. The only difference is in the scanning behaviour of the sensor. In this article, it is assumed that the sensor scans the whole surveillance region (attempting to measure each target in the region) before returning a "no detection" report. The measurement report generation process for a sensor is described in a flow chart like diagram in Fig. 2. The specific assumptions of the measurement model, that are made in this article, are listed below:

- Each sensor report contains at most one measurement, i.e. $\boldsymbol{Y}_k$ is either an empty set or a singleton $\{\underline{\boldsymbol{y}}_k\}$. The measurement $\underline{\boldsymbol{y}}_k$ is either a false alarm or a target-generated measurement.
- Each sensor report contains a false alarm with probability $p_f$. In addition, the false alarms are assumed to be distributed according to a false alarm density $f_f$ that is independent of the true target states.
- It is *a priori* unknown in which order a sensor scans its surveillance space.
- The sensor scans the whole surveillance space before reporting a "no detection".
- The probability of detection $p_d$ is constant, i.e. does not depend on target states.
- The model for a target-generated measurement is linear-Gaussian, $\underline{\boldsymbol{y}}_k \sim N(\mathrm{H}_k\underline{\boldsymbol{x}}_k, \mathrm{R}_k)$ for some constant matrices $\mathrm{H}_k$ and $\mathrm{R}_k$, that can depend on the time index $k$.

This kind of sensor model $P_{\boldsymbol{Y}_k|\boldsymbol{X}_k}$ can be characterised using an auxiliary random variable $\boldsymbol{c}_k$ taking values in $\mathbb{N} \cup \{\odot\}$, that determines the measurement to target *association*. The value $\odot$ denotes the case "no detection" and zero corresponds to a false alarm. Otherwise, the value of the association indicator tells which target caused the measurement. The prior distribution of the association variable depends only on the cardinality of $\boldsymbol{X}_k$, i.e. the current number of targets. The prior of $\boldsymbol{c}_k$ can be given as follows

$$P(\boldsymbol{c}_k = c \mid |\boldsymbol{X}_k| = n) = \begin{cases} p_f, & c = 0 \\ \dfrac{(1-p_f)(1-p_m)}{n}, & 1 \leq c \leq n \\ (1-p_f)p_m, & c = \odot \end{cases} \tag{10}$$

where $p_m = (1-p_d)^n$ is the probability that the sensor fails to produce any measurement. To answer the question

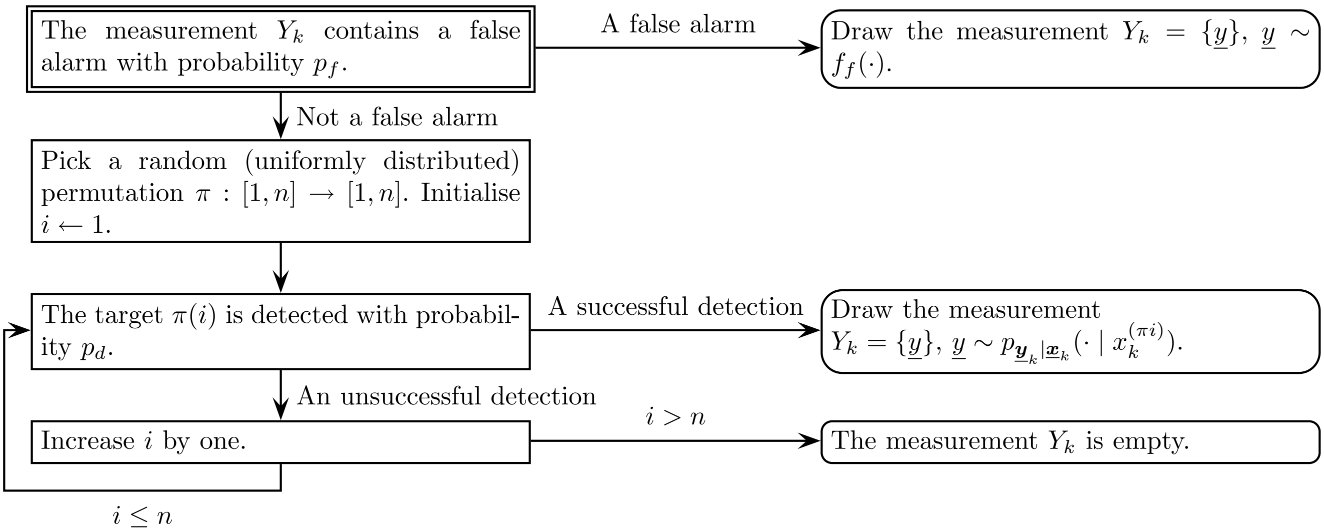



Fig. 2: The measurement report generation process of a sensor. The double framed box is the start point, and the rounded boxes are the end boxes determining the outcome of the measurement process.

"why" the distribution is defined as given in Eq. (10), one needs to see the measurement process depicted in Fig. 2. The first case, $c = 0$, is quite clear. It is rather easy to see also that when $c = \odot$, one needs a non-false alarm happening with probability $(1-p_f)$. In addition, one needs $n$ failed detections, the case happening with probability $(1-p_d)^n$ regardless of the random permutation (processing order) $\pi$. Assuming the detection process is independent of the false alarm process, one obtains the case $c = \odot$ in Eq. (10). Finally, the case $1 \leq c \leq n$ follows most easily reasoning "backwards": The conditional distribution of $\boldsymbol{c}_k$ must sum to unity, so the probability mass that is "left over" from the false alarm ($c = 0$) and "no detection" ($c = \odot$) cases is $(1-p_f)(1-p_m)$. Since the probability of detection was assumed uniform, and the processing order random, the only reasonable choice is to distribute the mass equally within the targets.

The conditional belief-mass function of the measurement $\boldsymbol{Y}_k$ can be given as follows

$$\beta_{\boldsymbol{Y}_k|\boldsymbol{X}_k,\boldsymbol{c}_k}(C \mid X, c) = \begin{cases} P_f(C), & c = 0 \\ P_{\boldsymbol{y}_k|\boldsymbol{x}_k}(C \mid x(c)), & c \geq 1 \\ 1, & c = \odot \end{cases}$$

where $x(c)$ denotes the $c$:th target picked from the set of targets $X$ (ordered in an arbitrary manner). The density of this belief-mass function can be given as follows (see the appendix for set derivation rules)

$$p_{\boldsymbol{Y}_k|\boldsymbol{X}_k,\boldsymbol{c}_k}(Y \mid X, c) = \begin{cases} f_f(y), & c = 0, Y = \{y\} \\ p_{\boldsymbol{y}_k|\boldsymbol{x}_k}(y \mid x(c)), & c \geq 1, Y = \{y\} \\ 1, & c = \odot, Y = \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Now, one can obtain the belief-mass function $\beta_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(C \mid X) = P(\boldsymbol{Y}_k \subset C \mid \boldsymbol{X}_k = X)$ as follows

$$\begin{aligned} \beta_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(C \mid X) &= \sum_c \beta_{\boldsymbol{Y}_k,\boldsymbol{c}_k|\boldsymbol{X}_k}(C, c \mid X) \\ &= \sum_c \left[\beta_{\boldsymbol{Y}_k|\boldsymbol{X}_k,\boldsymbol{c}_k}(C \mid X, c) P(\boldsymbol{c}_k = c \mid |\boldsymbol{X}_k| = n)\right] \end{aligned}$$

Likelihood of the multitarget state can be obtained as the set derivative of this belief-mass function and can be given as follows

$$\begin{aligned} p_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(\emptyset \mid X) &= (1-p_f)p_m \\ p_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(\{y\} \mid X) &= p_f f_f(y) \\ &\quad + \sum_{x \in X} \left[(1-p_f)\frac{1-p_m}{|X|} p_{\boldsymbol{y}_k|\boldsymbol{x}_k}(y \mid x)\right] \end{aligned}$$

where $y$ is the measurement, $X$ is the multitarget state, and the sum is defined to be zero if $X = \emptyset$. The symbols $p_d$ and $p_f$ correspond to the probability of detection and the probability of false alarm, respectively. Likewise, $p_{\boldsymbol{y}_k|\boldsymbol{x}_k}$ and $f_f$ correspond to the true measurement density and the false alarm density, respectively.

### C. Filtering

The random set estimation is just Bayesian recursive estimation, given in the general form as follows

$$\pi_{k|k-1}(X_k) = \int f_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}(X_k \mid X_{k-1}) \pi_{k-1|k-1}(X_{k-1})\delta X_{k-1} \tag{12}$$

$$\pi_{k|k}(X_k) = \frac{f_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(Y_k \mid X_k)\pi_{k|k-1}(X_k)}{\int f_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(Y_k \mid X'_k)\pi_{k|k-1}(X'_k)\delta X'_k} \tag{13}$$

where the notation $\pi_{a|b}(X) \triangleq p_{\boldsymbol{X}_a|\boldsymbol{Y}_{1:b}}(X \mid Y_{1:b})$ is used. Eq. (12) is referred to as the *prediction* and Eq. (13) the *measurement update* equation. This estimation problem is inherently nonlinear and often multimodal, and hence cannot be solved by conventional methods, such as the Kalman filter.

The model introduced above in Sections III-A and III-B is a special case of a model that can be decomposed into a discrete variable, and a tractable sub-model conditioned on the values of the discrete variable. When given the values of the following discrete variables, the rest of the model reduces into a linear-Gaussian one, in which filtering can be performed in closed form:

1) The variable $\boldsymbol{b}_k$ determining the number of born targets in each processing interval.
2) Association variable, $\boldsymbol{c}_k$, determining from which target (if any) the measurement $\boldsymbol{Y}_k$ originated.
3) Indicator variable $\boldsymbol{d}_k$ that determines whether each target disappears or survives in the interval $(t_{k-1}, t_k]$.

To summarise, the model can be reformulated to consist of random elements $\boldsymbol{z}_k = (\boldsymbol{X}_k, \boldsymbol{r}_k)$ where $\boldsymbol{r}_k = (\boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k)$. The conditional posterior measures $P_{\boldsymbol{X}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k}}(\cdot \mid r_{0:k}, Y_{1:k})$ can be obtained in closed form. Consequently, knowing the conditional posterior measures, one can obtain the conditional expectations $\mathsf{E}\left[h(\boldsymbol{X}_{0:k}, \boldsymbol{r}_{0:k}) \mid \boldsymbol{Y}_{1:k} = Y_{1:k}, \boldsymbol{r}_{0:k} = r^{(i)}_{0:k}\right]$ for functions of interest $h$, and compute the estimate given in Eq. (1). Using this information, Sect. IV continues to introduce Rao-Blackwellised particle filters for this model.

## IV. Rao-Blackwellised Random Set Particle Filters

This section describes three Rao-Blackwellised particle filters for the random set model described in Sect. III. RBPF is straightforward to apply to multitarget filtering given that the model is of the type described in Sect. III-C. All that is needed is to provide an importance distribution for sampling of the variables $\boldsymbol{r}_k = (\boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k)$, and computing the weights as given in Eq. (2).

This section describes three RBPF algorithms that differ only on definition of the importance distribution. The three importance distributions, $q^A_k$, $q^B_k$, and $q^C_k$, are defined differently. At first, Sect. IV-A describes the simplest RBPF implementation, using the predictive importance distribution for all the sampled variables: $(\boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k)$. Sect. IV-B introduces a bit more complex importance distribution employing optimal sampling[13] for $\boldsymbol{c}_k$, and finally Sect. IV-C describes an importance distribution providing almost optimal sampling of all the variables $(\boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k)$.

### A. RBPF A: Birth, Death, and Association Variables Drawn from the Predictive Distribution

The first RBPF algorithm is relatively simple. The discrete variables $(\boldsymbol{r}_i)_{i=1}^{\infty}$, where $\boldsymbol{r}_k = (\boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k)$ are sampled recursively from the predictive distribution. That is,

$$\begin{aligned} r^{(i)}_k \sim q^A_k(r^{(i)}_k \mid r^{(i)}_{0:k-1}, Y_{1:k}) &= p_{\boldsymbol{r}_k|\boldsymbol{r}_{0:k-1}}(r^{(i)}_k \mid r^{(i)}_{0:k-1}) \\ &= p_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1}}(c^{(i)}_k \mid b^{(i)}_k, d^{(i)}_k, r^{(i)}_{0:k-1}) \\ &\qquad p_{\boldsymbol{d}_k|\boldsymbol{r}_{0:k-1}}(d^{(i)}_k \mid r^{(i)}_{0:k-1}) p_{\boldsymbol{b}_k}(b^{(i)}_k) \end{aligned} \tag{14}$$

where $p_{\boldsymbol{b}_k}$ is the prior density of the number of born targets at time instant $t_k$ given in Eq. (5). The distribution of target deaths, $p_{\boldsymbol{d}_k|\boldsymbol{r}_{0:k-1}}$, is dependent only on the current number of targets (i.e. on the previous births and deaths $\boldsymbol{b}_{0:k-1}$ and $\boldsymbol{d}_{0:k-1}$). The association indicator distribution $p_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1}}$ given in Eq. (10), is dependent only on the number of targets at time instant $k$.

Substituting Eq. (14) to Eq. (2), the RBPF weight update formula reduces to

$$\begin{aligned} w^{(i)}_k &\propto w^{(i)}_{k-1} p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r^{(i)}_{0:k}, Y_{1:k-1}) \\ &= w^{(i)}_{k-1} \int p_{\boldsymbol{Y}_k,\boldsymbol{X}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(Y_k, X \mid r^{(i)}_{0:k}, Y_{1:k-1}) \delta X \\ &= w^{(i)}_{k-1} \int p_{\boldsymbol{Y}_k|\boldsymbol{c}_k,\boldsymbol{X}_k}(Y_k \mid c^{(i)}_k, X) \\ &\qquad p_{\boldsymbol{X}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(X \mid r^{(i)}_{0:k}, Y_{1:k-1}) \delta X \end{aligned} \tag{15}$$

In the case of a clutter association, i.e. $c^{(i)}_k = 0$, one has $p_{\boldsymbol{Y}_k|\boldsymbol{c}_k,\boldsymbol{X}_k} = f_f$ according to Eq. (11), so the integral in Eq. (15) reduces into

$$f_f(\underline{y}_k) \int p_{\boldsymbol{X}_k|\boldsymbol{r}^{(i)}_{0:k},\boldsymbol{Y}_{1:k-1}}(X \mid r^{(i)}_{0:k}, Y_{1:k-1}) \delta X = f_f(\underline{y}_k)$$

since the integral above equals to one. In the case $c^{(i)}_k \geq 1$, i.e. the measurement is associated to a true target, and the term $p_{\boldsymbol{Y}_k|\boldsymbol{c}_k,\boldsymbol{X}_k} = p_{\underline{\boldsymbol{y}}_k|\underline{\boldsymbol{x}}_k}$. Now, the integral in Eq. (15) can be given as follows

$$\begin{aligned} &\int \prod_{j=1}^{n_i} \left[ p_{\underline{\boldsymbol{x}}_k(j)|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}} \left( \underline{x}(j) \mid r^{(i)}_{0:k}, Y_{1:k-1} \right) \right] \\ &\qquad p_{\underline{\boldsymbol{y}}_k|\underline{\boldsymbol{x}}_k} \left( \underline{y}_k \mid \underline{x}(c^{(i)}_k) \right) \mathrm{d}\underline{x}(1) \cdots \mathrm{d}\underline{x}(n_i) \\ &= \int p_{\underline{\boldsymbol{x}}_k(c^{(i)}_k)|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}} \left( \underline{x}(c^{(i)}_k) \mid r^{(i)}_{0:k}, Y_{1:k-1} \right) \\ &\qquad p_{\underline{\boldsymbol{y}}_k|\underline{\boldsymbol{x}}_k} \left( \underline{y}_k \mid \underline{x}(c^{(i)}_k) \right) \mathrm{d}\underline{x}(c^{(i)}_k) \end{aligned} \tag{16}$$

where $n_i$ is the number of targets in the $i$:th particle, and $p_{\underline{\boldsymbol{x}}_k(c^{(i)}_k)|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(\cdot \mid r^{(i)}_{0:k}, Y_{1:k-1})$ is the predicted (prior) distribution of the $c^{(i)}_k$:th target. This integral is the *Kalman filter likelihood* and it can be computed in closed form [11].

The top-level procedure of the first RBPF algorithm is described in Alg. 1. The PREDICT procedure, described in Alg. 2, determines how samples $b^{(i)}_k$ and $d^{(i)}_k$ are drawn, and how the predicted sufficient statistics $T^{(i)}_-$ are obtained. The KF_PREDICT refers to Kalman filter prediction (see [11]), and the time difference of consecutive detection reports $\tau_k$ is supplied in order to compute the prediction matrices in Eq. (4). Poisson in Alg. 2 refers to the Poisson distribution, given in Eq. (6). Alg. 3 describes the procedure MEAS_UPDATE determining how

[13] In this article, "optimal sampling" refers to sampling from the posterior distribution (conditioned on the history of the particle).

$c_k^{(i)}$ are sampled, how the updated sufficient statistics $T_+^{(i)}$ are obtained, and how the weight update is performed. KF_UPDATE refers to the Kalman filter update and the Kalman filter likelihood [11]. The procedure VISUALISE refers to output and visualisation issues considered in Sect. V. The RESAMPLE procedure refers to resampling/selection phase of the particle filter.

```
// Initialise particles
T^(i) ← ∅                                          1 ≤ i ≤ N
w^(i) ← 1/N                                        1 ≤ i ≤ N
// Over the measurement report sequence
for k = 1 to ∞ do
  // Over the particles
  for i = 1 to N do
    (T_-^(i)) ← PREDICT(T^(i), τ_k)
    (T_+^(i), w̃_+^(i)) ← MEAS_UPDATE(T_-^(i), w^(i), Y_k)
  end for
  w_+^(i) ← w̃_+^(i) / Σ_{j=1}^N w̃_+^(j)             1 ≤ i ≤ N
  VISUALISE(T_+, w_+)
  (T, w) ← RESAMPLE(T_+, w_+)
end for
```

Alg. 1: Rao-Blackwellised particle filter: top-level procedure. The shorthand notations like $w \triangleq (w^{(1)}, \dots, w^{(N)})$ are used also for $T^{(i)}$ and $w_+^{(i)}$.

```
(T_-^(i)) ← PREDICT(T^(i), τ_k)

n ← |T^(i)|.
// Choose an arbitrary order for the elements
{(m_1, P_1), ..., (m_n, P_n)} ← T^(i).
T_-^(i) ← ∅
// Determine deaths and predict the single target states
for j = 1 to n do
  p̂_s ← 1 − ητ_k/n
  d_k^(i)(j) ∼ U(·; 0, 1)
  if d_k^(i)(j) ≤ p̂_s then
    (m_j^-, P_j^-) ← KF_PREDICT(m_j, P_j, τ_k)
    T_-^(i) ← T_-^(i) ∪ {(m_j^-, P_j^-)}
  end if
end for
// Augment the state by born targets
b_k^(i) ∼ Poisson(·; ητ_k)
for j = 0 to b_k^(i) do
  T_-^(i) ← T_-^(i) ∪ {(m_0, P_0)}
end for
```

Alg. 2: Rao-Blackwellised particle filter: prediction.

```
(T_+^(i), w_+^(i)) ← MEAS_UPDATE(T_-^(i), w_-^(i), Y_k)

n ← |T_-^(i)|
// Choose an arbitrary order for the elements
{(m_1, P_1), ..., (m_n, P_n)} ← T_-^(i).
if Y_k = {y_k} then
  // Measurement: form the importance distribution
  q̂_0 ← p_f
  q̂_j ← (1 − p_f)(1 − (1 − p_d)^n)/n               1 ≤ j ≤ n
  q_j ← q̂_j / Σ_{i=0}^n q̂_i                        0 ≤ j ≤ n
  c_k^(i) ∼ q_j(·)
  if c_k^(i) ≠ 0 then
    (m_{c_k^(i)}, P_{c_k^(i)}, L) ← KF_UPDATE(m_{c_k^(i)}, P_{c_k^(i)}, y_k)
  else
    L ← f_f(y_k)
  end if
else
  // No measurement: only update the particle weight
  L ← (1 − p_f)(1 − p_d)^n
end if
// Update the weight and store the updated particle state
w_+^(i) ← L w_-^(i)
T_+^(i) ← {(m_1, P_1), ..., (m_n, P_n)}
```

Alg. 3: Rao-Blackwellised particle filter: measurement update.

### *B. RBPF B: Predictive Birth and Death Sampling, Optimal Importance Distribution for the Association Variable*

The second RBPF algorithm is a straightforward extension of the first, RBPF A, described in Sect. IV-A. In this algorithm, the birth and death variables are sampled from the predictive prior distribution, just like in RBPF A. This implies that Alg. 2 applies directly also to this algorithm. However, the association variable is sampled from the optimal importance distribution as described in [15]. This means that the samples $r_k^{(i)}$ are drawn from

$$\begin{aligned} r_k^{(i)} &\sim q_k^B(r_k^{(i)} \mid r_{0:k-1}^{(i)}, Y_{1:k}) \\ &= q^B_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k}}(c_k^{(i)} \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k}) \\ &\qquad p_{\boldsymbol{d}_k|\boldsymbol{r}_{0:k-1}}(d_k^{(i)} \mid r_{0:k-1}^{(i)}) p_{\boldsymbol{b}_k}(b_k^{(i)}) \end{aligned} \tag{17}$$

where the importance distribution for the association variable can be given as follows

$$\begin{aligned} &q^B_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k}}(c_k^{(i)} \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k}) \\ &\quad = p_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k}}(c_k^{(i)} \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k}) \\ &= \frac{p_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1}}(c_k^{(i)} \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)})}{p_{\boldsymbol{Y}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k-1})} \\ &\qquad p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k}^{(i)}, Y_{1:k-1}) \end{aligned} \tag{18}$$

where $p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k}^{(i)}, Y_{1:k-1})$ is given by Eq. (15).

The weight update in Eq. (2) can be given in this case as follows

$$w_k^{(i)} \propto w_{k-1}^{(i)} \; p_{\boldsymbol{Y}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k-1})$$

where $p_{\boldsymbol{Y}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k-1})$ can be obtained from Eq. (18), since $\int p_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k}}(c \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k})\mathrm{d}c = 1$. The association variable $\boldsymbol{c}_k$ has a finite number of possible values, so the integration is, in fact, a finite sum, so the proportionality constant can be computed as follows:

$$\begin{aligned} &p_{\boldsymbol{Y}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}, Y_{1:k-1}) \\ &\quad = \sum_c \Big[ p_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1}}(c \mid b_k^{(i)}, d_k^{(i)}, r_{0:k-1}^{(i)}) \\ &p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{c}_k,\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k-1}^{(i)}, b_k^{(i)}, d_k^{(i)}, c, Y_{1:k-1}) \Big] \end{aligned} \tag{19}$$

In practice this means that when obtaining the importance distribution $q^B_{\boldsymbol{c}_k|\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k}}$ for the association variable $\boldsymbol{c}_k$, one obtains also the weight update factor. Alg. 4 summarises this measurement update procedure. The RBPF B algorithm is similar to RBPF A explained in Alg. 1, but MEAS_UPDATE is replaced with OPT_MEAS_UPDATE in Alg. 4.

### C. RBPF C: Approximately Optimal Importance Distribution for Birth, Death, and Association

Sect. IV-B showed how the importance distribution can be modified by sampling the association variable $\boldsymbol{c}_k$ from an optimal importance distribution given the values for the birth ($\boldsymbol{b}_k$) and death ($\boldsymbol{d}_k$) variables. It is a tempting idea to create an optimal importance distribution, from which all $(\boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k) = \boldsymbol{r}_k$ can be sampled. Theoretically it is straightforward, just set

$$\begin{aligned} r_k^{(i)} &\sim q_k^C(r_k^{(i)} \mid r_{0:k-1}^{(i)}, Y_{1:k}) \\ &= p_{\boldsymbol{r}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k}}(r_k^{(i)} \mid r_{0:k-1}^{(i)}, Y_{1:k}) \\ &= \frac{p_{\boldsymbol{r}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(r_k^{(i)} \mid r_{0:k-1}^{(i)}, Y_{1:k-1})}{p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k-1}^{(i)}, Y_{1:k-1})} \\ &\qquad p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k}^{(i)}, Y_{1:k-1}) \end{aligned} \tag{20}$$

Again, as in RBPF B, the variable $\boldsymbol{r}_k$ is discrete, but unlike in RBPF B, there are infinite feasible realisations of $\boldsymbol{r}_k$. Hence, $q_k^C$ must be approximated in order to obtain a practical importance distribution. The approximation that is chosen is based on two simplifying assumptions:

- At most one target can appear during a processing interval.
- At most one target can disappear during a processing interval.

In fact, the restriction of at most one born target per processing interval is enough to guarantee that $\boldsymbol{r}_k$ has only a finite number of possibilities. The restriction that at most one target can disappear is introduced in order to keep the number of combinations, and hence the complexity of the algorithm, practical.

---

$(T_+^{(i)}, w_+^{(i)}) \leftarrow$ OPT_MEAS_UPDATE$(T_-^{(i)}, w_-^{(i)}, Y_k)$

$n \leftarrow |T_-^{(i)}|$
*// Choose an arbitrary order for the elements*
$\{(\underline{m}_1^-, \mathrm{P}_1^-), \ldots, (\underline{m}_n^-, \mathrm{P}_n^-)\} \leftarrow T_-^{(i)}$.
$J \leftarrow \emptyset$
**if** $Y_k = \{\underline{y}_k\}$ **then**
  *// Measurement: perform the Kalman update*
  **for** $j = 1$ to $n$ **do**
    $(\underline{m}_j^+, \mathrm{P}_j^+, L) \leftarrow$ KF_UPDATE$(\underline{m}_j^-, \mathrm{P}_j^-, \underline{y}_k)$
    $\widehat{q}_j \leftarrow L(1-p_f)(1-(1-p_d)^n)/n$
  **end for**
  *// Form the optimal importance distribution for* $\boldsymbol{c}_k$
  $\widehat{q}_0 \leftarrow p_f f_f(\underline{y}_k)$
  $\Lambda \leftarrow \sum_{i=0}^n \widehat{q}_i$
  $q_j \leftarrow \widehat{q}_j/\Lambda$   $0 \le j \le n$
  $c_k^{(i)} \sim q_j(\cdot)$
  **if** $c_k^{(i)} \neq 0$ **then**
    $J \leftarrow \{c_k^{(i)}\}$
  **end if**
**else**
  *// No measurement: only update the particle weight*
  $\Lambda \leftarrow (1-p_f)(1-p_d)^n$
**end if**
*// Update the weight and store the updated particle state*
$w_+^{(i)} \leftarrow \Lambda w_-^{(i)}$
$T_+^{(i)} \leftarrow \{j \in J : (\underline{m}_j^+, \mathrm{P}_j^+)\} \cup \{j \in J^- : (\underline{m}_j^-, \mathrm{P}_j^-)\}$, where $J^- = \{1, \ldots, n\} \setminus J$.

---

Alg. 4: Rao-Blackwellised particle filter: measurement update with optimal measurement association importance distribution.

The approximation that no more than one target is born can be achieved easily just by truncating the Poisson birth distribution, i.e. setting $b_j = 0$ for $j \geq 2$ in Eq. (6). The death distribution can be truncated as well, so that any event containing more than one death is assigned probability zero[14]. It must be noted that after truncating the distributions, the constraint of Eq. (7) is not fulfilled, and strictly speaking, this importance distribution obtained by truncation is not even valid, i.e. does not cover the whole support of the posterior distribution. However, with small $\eta\tau_k$, the probabilities assigned to zero in truncation are already small, and the error made is small.

[14] In fact, the truncated "distributions" are not proper probability distributions, but they can be normalised to such by multiplying with a constant.

The weight update formula can be given as follows

$$w_k^{(i)} \propto w_{k-1}^{(i)} p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k-1}^{(i)}, Y_{1:k-1})$$

and one can use the results derived for RBPF B, since

$$\begin{aligned} & p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k-1}^{(i)}, Y_{1:k-1}) \\ &= \iint p_{\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{Y}_{1:k-1}}(b,d,Y_k \mid r_{0:k-1}^{(i)}, Y_{1:k-1}) \mathrm{d}b\mathrm{d}d \\ &= \sum_{b,d} \Big[ p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k-1}^{(i)}, b, d, Y_{1:k-1}) \\ &\qquad p_{\boldsymbol{b}_k}(b) p_{\boldsymbol{d}_k|\boldsymbol{r}_{0:k-1}}(d \mid r_{0:k-1}^{(i)}) \Big] \end{aligned}$$

where $p_{\boldsymbol{Y}_k|\boldsymbol{r}_{0:k-1},\boldsymbol{b}_k,\boldsymbol{d}_k,\boldsymbol{Y}_{1:k-1}}(Y_k \mid r_{0:k-1}^{(i)}, b, d, Y_{1:k-1})$ is obtained as given in Eq. (19).

The RBPF algorithm using this importance distribution is similar to Alg. 1, but with both PREDICT and MEAS_UPDATE procedures replaced by OPT_UPDATE procedure summarised in Alg. 5. The formation of the approximated importance distribution $q_k^C$ is described separately in Alg. 6.

```
(T_+^(i), w_+^(i)) ← OPT_UPDATE(T^(i), w^(i), τ_k, Y_k)

n ← |T^(i)|.
// Choose an arbitrary order for the elements
{(m_1, P_1), ..., (m_n, P_n)} ← T^(i).
for j = 1 to n do
  (m_j^-, P_j^-) ← KF_PREDICT(m_j, P_j, τ_k)
end for
if Y_k = {y_k} then
  // Update the state of a born target
  (m_0, P_0, L_B) ← KF_UPDATE(m_0, P_0, y_k)
  for j = 1 to n do
    (m_j^+, P_j^+, L_j) ← KF_UPDATE(m_j^-, P_j^-, y_k)
  end for
else
  L_B ← L_1 ← ··· ← L_n ← 0
end if
(J, D, b_k^(i), Λ) ← IMPORTANCE(τ_k, L_B, (L_j)_{j=1}^n, Y_k)
// Update the weight and store the updated particle state
w_+^(i) ← Λ w^(i)
// The updated target, if any
T_+^(i) ← {j ∈ J : (m_j^+, P_j^+)}
// The targets neither updated nor died
T_+^(i) ← T_+^(i) ∪ {j ∈ {1,...,n} \ (J ∪ D) : (m_j^-, P_j^-)}
// The born targets
if b_k^(i) = 1 then
  T_+^(i) ← T_+^(i) ∪ {(m_0, P_0)}
end if
```

Alg. 5: Rao-Blackwellised particle filter: update with almost optimal importance distribution. The function IMPORTANCE refers to Alg. 6.

```
(J, D, b, Λ) ← IMPORTANCE(τ_k, L_B, (L_j)_{j=1}^n, Y_k)

p̂_s ← { 1,             n = 0
       { 1 − ητ_k/n,   n > 0
// Form the death indicator distribution.
p̂_D(0) ← p̂_s^n
p̂_D(j) ← p̂_s^{n−1}(1 − p̂_s)                       1 ≤ j ≤ n
p_D(j) ← p̂_D(j) / Σ_i p̂_D(i)                       0 ≤ j ≤ n
// Form the association prior p_C(j, n') and the
// association likelihoods L̂.
if Y_k = ∅ then
  p_C(j, n') ← (1 − p_f)(1 − p_d)^{n'}             0 ≤ n' ≤ n + 1
                                                   0 ≤ j ≤ n + 1
  L̂_0 ← 1
  L̂_j ← 0                                          1 ≤ j ≤ n + 1
else
  p_C(0, n') ← p_f                                 0 ≤ n' ≤ n + 1
  p_C(j, 0) ← 0                                    1 ≤ j ≤ n + 1
  p_C(j, n') ← (1 − p_f) (1 − (1 − p_d)^{n'}) / n'  1 ≤ n' ≤ n + 1
                                                   1 ≤ j ≤ n + 1
  L̂_0 ← f_f(y_k)
  L̂_j ← L_j                                        1 ≤ j ≤ n
  L̂_{n+1} ← L_B
end if
// Form the importance distribution.
q̂(j, k, ℓ) = 0 for all j, k, ℓ
for k = 0 to 1 do
  for ℓ = 0 to n do
    n' ← n + k − ℓ' where ℓ' = 1 if ℓ ≠ 0
    for j = 0 to n + k except ℓ if ℓ ≠ 0 do
      q̂(j, k, ℓ) ← Poisson(k; ητ_k) p_D(ℓ) p_C(j, n') L̂_j
    end for
  end for
end for
Λ ← Σ_{j,k,ℓ} q̂(j, k, ℓ)
q(j, k, ℓ) ← q̂(j, k, ℓ)/Λ
J ← ∅, D ← ∅
(c, b, d) ∼ q(·, ·, ·)
if c ≠ 0 then
  J ← {c}
end if
if d ≠ 0 then
  D ← {d}
end if
```

Alg. 6: Sampling from the almost optimal importance distribution.

## V. Output and Visualisation

The RBPF algorithms presented in Sect. IV maintain a representation of the random set posterior distribution. The posterior distribution approximation, as such, is of little interest. One needs to extract relevant *estimators* and/or *visualise* the distribution in some manner. The Joint and Marginal Multitarget Estimators (JoME and

MaME) proposed by Mahler [25] cannot be obtained in closed form in the case of a RBPF posterior distribution, due to multimodality of the distribution.

The Expected A Posteriori (EAP) and Maximum A Posteriori (MAP) estimates for the number of targets in the surveillance space, however, can be obtained relatively straightforwardly [5],

$$\widehat{n}^{\mathrm{EAP}} = \sum_{j=0}^{\infty} \Big[ j\widehat{p}(j) \Big], \qquad \widehat{n}^{\mathrm{MAP}} = \arg\max_{j\in\mathbb{N}} \widehat{p}(j) \quad (21)$$

where

$$\widehat{p}(j) = \sum_{\{i:|T^{(i)}|=j\}} w^{(i)}$$

and $w^{(i)}$ are the particle weights as denoted in Sect. IV.

Since the position distribution of the targets is even more important than the cardinality distribution, it is clear that the position distribution must be illustrated in some manner. The Probability Hypothesis Density (PHD) provides a "histogram like" visualisation of the multitarget distribution.[15] In [5] it is shown how a PHD visualisation is applied in the case of a bootstrap particle filter. The same idea, dividing the surveillance space into finite-resolution cells, does not work with the RBPF approach, since it would lead into computation of infeasible integrals. In the case of RBPF, one can take advantage of the continuous nature of the estimated distribution, and obtain the PHD as follows (see the appendix)

$$D(\underline{x}) = \sum_{i=1}^{N} w^{(i)} \sum_{j=1}^{n_i} p_N(\underline{x}; \underline{m}_j^{(i)}, \mathrm{C}_j^{(i)}) \quad (22)$$

where $n_i$ is the number of targets in $i$:th particle, $(\underline{m}_j^{(i)}, \mathrm{C}_j^{(i)}) \in T^{(i)}$ are the mean and covariance of $j$:the target in $i$:th particle, and $p_N(\cdot; \underline{m}, \mathrm{C})$ denotes the density of normal distribution $N(\cdot; \underline{m}, \mathrm{C})$.

The PHD visualisation within the RBPF framework is straightforward, but computationally expensive, since evaluation of the PHD requires evaluation of the value of all the Gaussian densities in the posterior approximation. The PHD visualisation is, in addition, problematic in the sense that it discards any inter-target dependencies. It is hence important to develop other visualisation methods and to develop algorithms computing relevant point estimates, and most importantly approximating the JoME and MaME estimators. These issues are out of the scope of this article, and are postponed to future research. Some ideas regarding these issues are outlined below.

The RBPF multitarget tracker employs a similar data structure to the Multiple Hypothesis Tracking (MHT) algorithm [16], since the discrete variables $r_{0:k}^{(i)}$ essentially form a *hypothesis* of association, birth, and death of the targets. Therefore, one could extract a "winner particle" (corresponding to a "winner hypothesis" in MHT) in each iteration. Given this "winner particle", it is easy to show

[15] PHD is defined to be the density of the measure $A \mapsto \mathsf{E}[|\boldsymbol{X} \cap A|]$, where $A$ is a measurable set, and $\boldsymbol{X}$ is a random set. That is, if $D : \mathbb{R}^d \to [0,\infty)$ denotes the PHD, one obtains $\mathsf{E}[|\boldsymbol{X} \cap A|] = \int_A D(\underline{x})\mathrm{d}\underline{x}$.

the estimated target states (means) and possibly their uncertainties (covariance ellipsoids). So, how to extract this "winner particle"?

$$\widehat{j} = \arg\max_{1\le i\le N} \widehat{w}^{(i)} \quad \text{where} \quad \widehat{w}^{(i)} = \sum_{\ell=1}^{N} w^{(\ell)} \delta_{r_{0:k}^{(i)}}(r_{0:k}^{(\ell)})$$

The term $\widehat{w}^{(i)}$ is just the weight $w^{(i)}$, but the possible duplicate particles, i.e. $r_{0:k}^{(i)} = r_{0:k}^{(\ell)}$, are taken into account. It must be noted that in general the maximum is not unique. In addition, since the obtained "winner particle" is just a "current best guess" of the birth, death, and association variables, it may well be misleading, and the uncertainties are generally too optimistic. Furthermore, it is obvious that the "best guess" can change dramatically after processing the next measurement.

The next idea is to consider some kind of a clustering framework, in which each target hypothesis in each particle would be first clustered, and then according to this clustering, the weighted mean and covariance estimate would be computed through moment matching. There are several open questions regarding the clustering to be answered, though:

- Should the clustering framework take into account only the particles (hypotheses) containing exactly $\widehat{n}^{\mathrm{MAP}}$ particles? The alternative is to take all the particles into account.
- Should the clustering framework use the inter-target dependence information, for example by not allowing two targets in a particle to be clustered into a same cluster?
- Should the clustering framework be time-dependent, at least in the sense that the clusters of the previous iteration would be used as a starting point for clustering in current iteration?

Such clustering algorithms could be developed in order to approximate JoME or MaME estimators.

## VI. Experimental Setup

When testing a multitarget tracking algorithm, there are numerous parameters, that can be changed (and *are* changed in a real-world application). Only few parameters can be altered in a limited test setup. In this article, two parameters were selected to be altered: the False Alarm Rate (FAR), and the number of particles. There were two test setups: an ideal scenario and a bit more practical tracking scenario. The first test setup in Sect. VI-A consists of the ideal situation, where both the single-target dynamic model and measurement model were linear-Gaussian. Both the tracks and the measurements in the scenario were generated randomly according to the models. This test is intended to test the efficiency of the algorithms described in Sect. IV as statistical estimators.

The second test setup in Sect. VI-B consists of a challenging bearings-only tracking scenario. The measurement model is not linear, but local linearisation i.e. the extended Kalman filter (EKF) is employed to enable the use of the

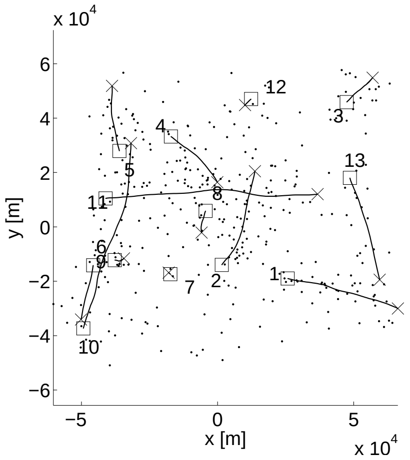



Fig. 3: Trajectories of tracks and $p_f = 0.2$ measurements in an example scenario generated according to the multitarget dynamic model. The tracks start from a square (□) and end in a cross (×). Measurements are depicted as dots (·).

RBPF tracking algorithm[16] [11]. In addition, the tracks were predefined manually, and hence did not obey the defined dynamic models. This test was considered a "proof-of-concept" showing the potential of the algorithms for practical application.

### A. Conditionally Linear-Gaussian Model

The kinematic model employed for the tests was the constant velocity model described in Sect. III-A.1 with process noise parameter value $\rho = 35\ \mathrm{m/s^{3/2}}$. The birth rate parameter and the probability of detection parameter had constant values, $\eta = 0.02$ and $p_d = 0.9$, respectively. The birth distribution was Gaussian with constant mean and covariance parameters $P_b = N(\underline{m}_0, \mathrm{C}_0)$ where $\underline{m}_0 = \underline{0}$ and $\mathrm{C}_0 = \mathrm{diag}(v_p, v_p, v_v, v_v)$ with position variance $v_p = 8\frac{1}{3} \times 10^8\ \mathrm{m}^2$ and velocity variance $v_v = \frac{1}{3} \times 10^6\ (\mathrm{m/s})^2$. The prior distribution of the initial random set state was set to $P(\boldsymbol{X}_0 = \emptyset) = 1$.

The test data consisted of synthetic, random tracks, which were generated according to the multitarget dynamic model described in Sect. III-A with the parameter values given above. There were 100 scenarios, each 600 seconds long, with measurements arriving every second. Fig. 3 and Fig. 4 show an example of such a randomly generated scenario.

The sensor model was linear-Gaussian, with measurement matrix parameter $\mathrm{H}_k = \left[\mathrm{I}_2, 0_2\right]$, where $\mathrm{I}_2$ and $0_2$ are two dimensional identity and zero matrices, respectively. The measurement noise matrix was $\mathrm{R}_k = \mathrm{diag}(\sigma_m^2, \sigma_m^2)$ where $\sigma_m = 500$ m is the standard deviation of the both $x$ and $y$ components. Two FARs, $p_f = 0.2$ and $p_f = 0.5$, were used in the tests. The false alarm distribution was

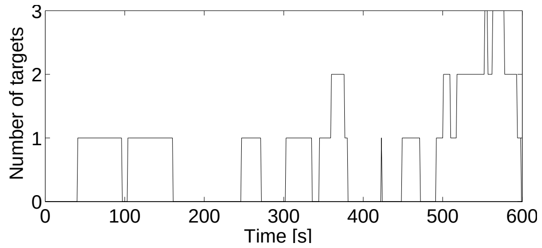



Fig. 4: Number of targets in the example scenario with respect to time.

uniform,

$$f_f(\underline{y}) = f_U(\underline{y}_x; -r, r) \times f_U(\underline{y}_y; -r, r)$$

where $r = 50$ km and $f_U$ corresponds to the density of a uniform distribution.

The RBPF algorithms were run with $N = 1000$, $N = 100$, and $N = 10$ particles. The resampling threshold of the adaptive resampling was one fourth of the number of particles. Table I shows the average errors of the three RBPF algorithms over 100 randomly generated target scenarios with two different FARs, $p_f = 0.2$ and $p_f = 0.5$. EAP and MAP refer to the estimators given in Eq. (21), and RMS and MAE correspond to the root mean square error and the mean average error of the corresponding estimators with respect to the true number of targets. It seems that there is a noticeable difference between RBPF A and the other two algorithms. On the other hand, RBPF B and RBPF C seem to produce almost comparable results with the same number of particles. RBPF A with 1000 particles seems to produce similar results to RBPF B and RBPF C with 100 particles. Likewise, RBPF A with 100 particles compares to RBPF B and RBPF C with 10 particles. This suggests that using the optimal association importance distribution in RBPF B and C reduces the variance significantly compared to the predictive association importance distribution in RBPF A. However, there seems to be little difference between the performance of RBPF B and RBPF C. This means that sampling births and deaths from the almost optimal importance distribution instead of the predictive distribution did not give significant advantage in this case.

Table II shows the results obtained using the direct particle filter implementation without Rao-Blackwellisation, described in [4], [5]. Comparing the results of the bootstrap implementation with 1000 particles and the RBPF implementation with 10 particles, only the RBPF A has worse results than the direct implementation. So, at least in this case, one can state that RBPF B and C "need" less than one percent of the sample size in the direct particle implementation in order to obtain the same accuracy.

### B. Bearings-only Tracking

In the bearings-only tracking scenario, similar models to the conditionally linear-Gaussian scenario were used. The

[16]Strictly speaking, since EKF is an approximation and not a closed-form solution, the algorithm is not *Rao-Blackwellised*. However, the idea is exactly the same.

TABLE I: Average errors of the RBPF algorithms over 100 linear-Gaussian scenarios.

| | | $p_f = 0.2$ | | | | | | | | | $p_f = 0.5$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $N = 10$ | | | $N = 100$ | | | $N = 1000$ | | | $N = 10$ | | | $N = 100$ | | | $N = 1000$ | | |
| | | A | B | C | A | B | C | A | B | C | A | B | C | A | B | C | A | B | C |
| EAP | RMS | 2.433 | 1.562 | 1.589 | 1.338 | 0.905 | 0.884 | 0.887 | 0.734 | 0.724 | 2.622 | 1.864 | 1.754 | 1.765 | 1.452 | 1.491 | 1.515 | 1.382 | 1.294 |
| | MAE | 1.746 | 1.077 | 1.076 | 0.898 | 0.582 | 0.569 | 0.571 | 0.473 | 0.470 | 2.030 | 1.369 | 1.264 | 1.289 | 1.044 | 1.087 | 1.108 | 1.025 | 0.948 |
| MAP | RMS | 2.314 | 1.656 | 1.538 | 1.456 | 0.992 | 0.956 | 0.976 | 0.783 | 0.781 | 2.559 | 1.905 | 1.815 | 1.829 | 1.550 | 1.592 | 1.627 | 1.496 | 1.415 |
| | MAE | 1.662 | 1.093 | 0.994 | 0.928 | 0.581 | 0.539 | 0.566 | 0.439 | 0.434 | 1.965 | 1.360 | 1.263 | 1.286 | 1.057 | 1.095 | 1.131 | 1.044 | 0.964 |

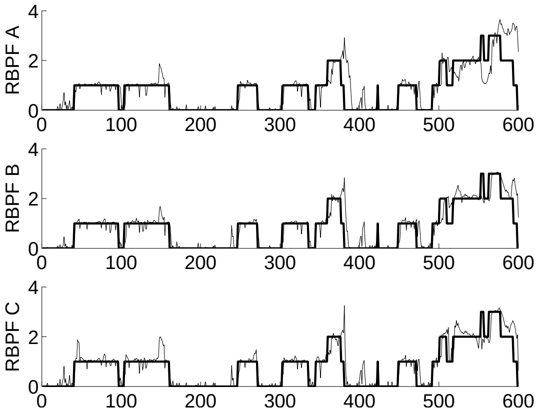



Fig. 5: EAP estimates of the number of targets with the RBPF algorithms run with 100 particles in the example scenario. The true number of targets is denoted with thick line.

TABLE II: Average errors of the bootstrap particle filter algorithm with 1000 particles over 100 randomly generated scenarios.

| | EAP | | MAP | |
|---|---|---|---|---|
| PFA | RMS | MAE | RMS | MAE |
| 0.2 | 1.8639 | 1.2640 | 1.8478 | 1.2359 |
| 0.5 | 1.9334 | 1.4669 | 2.1189 | 1.5666 |

measurement model was, of course, different, being

$$\boldsymbol{y} = \text{atan2}(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{x}}) + \boldsymbol{v} \mod 2\pi$$

where $\boldsymbol{v}$ is a Gaussian distributed random variable having standard deviation $\sigma_a = (3/180)\pi$, $\widehat{\boldsymbol{x}} = \boldsymbol{x}_x - s_x$ and $\widehat{\boldsymbol{y}} = \boldsymbol{x}_y - s_y$ are the target $x$ and $y$ coordinates with respect to the sensor position $\underline{s} = [s_x, s_y]^T$, and the function atan2 is the four quadrant inverse tangent.

The tracks and the sensors used in the bearings-only scenario are depicted in Fig. 6. The characteristics of the tracks are given in Table III. The tests consisted of 100 different sets of measurements generated from these tracks according to the sensor model. Each set of measurements contained 300 detection reports, generated from time 1 to 300, one measurement per second. The sensor reporting the measurement at each time instant was selected randomly.

TABLE III: The tracks used in the bearings-only tracking scenario.

| ID | Start time [s] | End time [s] | Speed [m/s] |
|---|---|---|---|
| 1 | 5 | 220 | 416.0 |
| 2 | 35 | 270 | 891.1 |
| 3 | 70 | 210 | 339.7 |

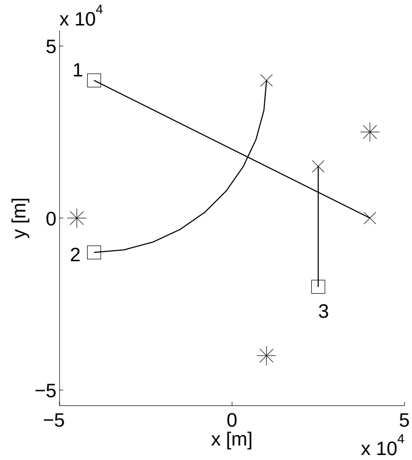



Fig. 6: Trajectories of the tracks in the bearings-only tracking scenario, and the sensor positions (✻).

The average errors of the different RBPF algorithms are shown in Table IV for FAR 0.2 and 0.5. Fig. 7 shows an example how the EAP estimates of the number of targets developed with the different RBPF algorithms. The algorithms were run with 1000 particles and FAR was set to value 0.2. It seems that RBPF A produced rather unreliable estimates in this case when the number of targets was three, whereas RBPF B and C managed the situation better. Fig. 8 shows snapshots of the PHD. The darker a pixel in the image, the larger the PHD value is[17]. The PHD was evaluated as given in Eq. (22), at one time instant from the scenario depicted in Fig. 6. The PHD images show that RBPF A seems to have "lost" one of the targets, and has some false target hypotheses near the upmost sensor. RBPF B and C have captured all the tracks, but it is clearly seen that their position uncertainty

[17]The intensities of the PHD images are scaled so that each image has one black pixel, and the other intensities are scaled linearly so that the value zero is white.

TABLE IV: Average errors of the RBPF algorithms over 3-target bearings-only scenario.

| | | $p_f = 0.2$ | | | | | | | | | $p_f = 0.5$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $N = 10$ | | | $N = 100$ | | | $N = 1000$ | | | $N = 10$ | | | $N = 100$ | | | $N = 1000$ | | |
| | | A | B | C | A | B | C | A | B | C | A | B | C | A | B | C | A | B | C |
| EAP | RMS | 3.046 | 2.702 | 2.215 | 2.046 | 0.922 | 0.853 | 0.703 | 0.544 | 0.540 | 2.509 | 2.493 | 2.494 | 2.139 | 1.551 | 1.541 | 1.162 | 0.824 | 0.823 |
| | MAE | 2.397 | 2.157 | 1.733 | 1.535 | 0.665 | 0.608 | 0.482 | 0.374 | 0.373 | 1.970 | 1.967 | 1.924 | 1.604 | 1.170 | 1.153 | 0.834 | 0.602 | 0.600 |
| MAP | RMS | 2.927 | 2.648 | 2.221 | 2.061 | 0.978 | 0.905 | 0.779 | 0.584 | 0.586 | 2.462 | 2.520 | 2.506 | 2.191 | 1.621 | 1.617 | 1.295 | 0.885 | 0.895 |
| | MAE | 2.283 | 2.076 | 1.693 | 1.505 | 0.615 | 0.557 | 0.445 | 0.294 | 0.298 | 1.909 | 1.958 | 1.891 | 1.604 | 1.134 | 1.130 | 0.847 | 0.552 | 0.552 |

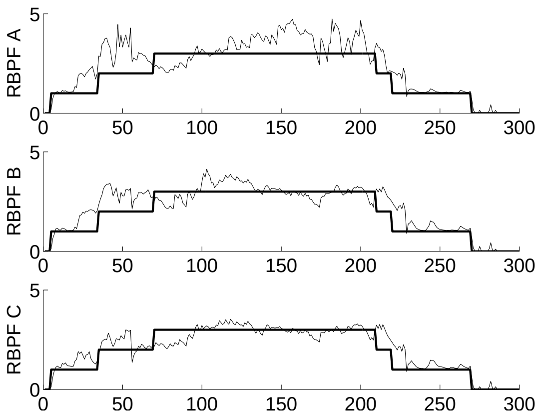



Fig. 7: EAP estimates of the number of targets with the RBPF algorithms run with 1000 particles in a bearings-only scenario. The true number of targets is denoted with thick line.

TABLE V: Average errors of the bootstrap particle filter algorithm with 1000 particles over bearings-only scenarios.

| | EAP | | MAP | |
|---|---|---|---|---|
| PFA | RMS | MAE | RMS | MAE |
| 0.2 | 6.0617 | 4.8298 | 4.6762 | 3.7707 |
| 0.5 | 3.5247 | 2.6805 | 3.3492 | 2.5155 |

is relatively large.

## VII. Conclusions and Future Work

This article introduced Rao-Blackwellised particle filters in FISST multitarget tracking. The RBPF FISST algorithms were derived for a class of multitarget models having a special sensor model, assuming that each detection report either contains one single-target measurement, or is a "no detection" report. The multitarget dynamic model that was proposed is a standard one, consisting of models for independent single-target movements, births, and deaths. However, a new "symmetrisation" of births and deaths was introduced in order to minimise the bias in the estimation of the number of targets.

The proposed approach was tested in two tracking scenarios: an ideal conditionally linear-Gaussian (CLG) scenario, and a challenging nonlinear bearings-only scenario. The results of the CLG scenario showed that RBPF provides a significant increase of the tracking efficiency (or reduction of variance of the estimate) when compared to a direct particle filter implementation without Rao-Blackwellisation. Based on the experiments, the Rao-Blackwellisation can reduce the number of particles needed for a given accuracy to less than one percent of the number needed by direct particle implementations. The RBPF algorithms were shown to be applicable also to a non-Gaussian model in the bearings-only tracking scenario. The results obtained from the bearings-only scenario are promising considering the practical capabilities of the approach.

The extension of the RBPF approach to a nonlinear measurement model in the bearings-only scenario was carried out within the EKF framework in this article. Similarly, one could employ UKF [12] or any other approximation approach. In addition to allowing nonlinear and non-Gaussian single-target dynamic models and measurement models, the probability of detection model could be allowed to be nonuniform. One of the future research topics is to survey and develop methods for extending the class of models to which the RBPF algorithms can be applied. Another important future research topic is the computation of descriptive point estimates, such as JoME and MaME [25], from the RBPF multitarget posterior distribution.

The presented RBPF algorithms are conceptually simpler than similar approaches, such as the MHT approach [13], [16], since no specific bookkeeping system and pruning heuristics are required. In addition, the method is theoretically justified, since the computed estimates converge to the true values (computed from the true multitarget posterior distribution) as the number of samples is increased. Comparison of the practical performance of the RBPF FISST approach with respect to the PHD filter and traditional multitarget tracking approaches, such as MHT and JPDA, is an important topic for future research.

## Acknowledgements

This work was carried out at and funded by Datactica Ltd. The author is most grateful to M.Sc. Kari Heine for careful review of the manuscript and to Dr. Tech. Petri Salmela at Datactica Ltd for support and constructive comments on the work.

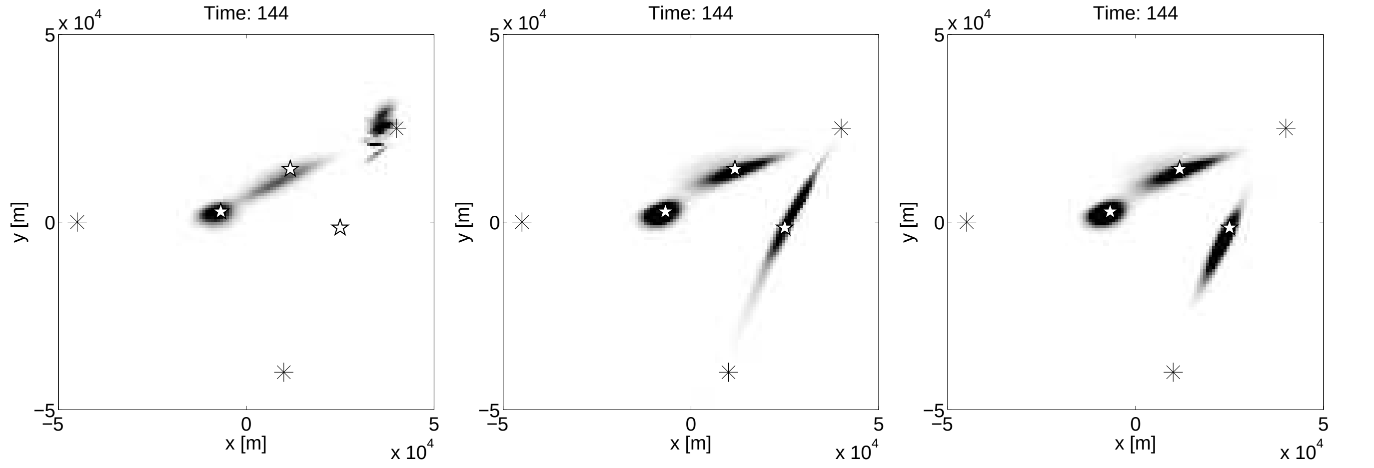



Fig. 8: PHD estimate of the different RBPF algorithms with 1000 particles evaluated at the same time instant in the bearings-only scenario. The star symbols (☆) denote the true target positions.

## Appendix

This appendix contains the proof of Eq. (22), i.e. the PHD of the RBPF posterior distribution. For convenience, some general properties of the set derivative are listed below [1]:

$$\frac{\delta}{\delta X}\left[a_1\beta_1(C)+a_2\beta_2(C)\right]=a_1\frac{\delta\beta_1}{\delta X}+a_2\frac{\delta\beta_2}{\delta X} \tag{23}$$

$$\frac{\delta}{\delta x}\beta_1(C)\beta_2(C)=\beta_1\frac{\delta\beta_2}{\delta x}+\beta_2\frac{\delta\beta_1}{\delta x} \tag{24}$$

$$\frac{\delta}{\delta X}P(C)^n=\begin{cases}\frac{n!}{(n-k)!}P(C)^{n-k}\prod_{i=1}^k f(x_i), & k\le n\\ 0, & k>n\end{cases} \tag{25}$$

where $P$ is a probability measure on $\mathbb{R}^n$, $f$ is the corresponding density, and $X=\{x_1,\ldots,x_k\}$. In addition, the following result is needed:

$$\left[\frac{\delta}{\delta z}\prod_{i=1}^n\beta_i\right](C)=\sum_{i=1}^n\left[\frac{\delta\beta_i}{\delta z}(C)\prod_{j\ne i}\beta_j(C)\right] \tag{26}$$

For proof, consult [4, p. 59]. The PHD can be obtained from the belief-mass function as follows [1, p. 169]

$$D_{\boldsymbol{X}}(\underline{x})=\frac{\delta\beta_{\boldsymbol{X}}}{\delta\underline{x}}(\mathbb{R}^d)$$

The posterior random set belief-mass function estimated by the RBPF algorithms can be written as follows

$$\widehat{\beta}_{\boldsymbol{X}_k|\boldsymbol{Y}_{1:k}}(C)=\sum_{i=1}^N w^{(i)}\beta_i(C) \tag{27}$$

where $C\subset\mathbb{R}^d$, and $\beta_i(C)\triangleq\beta_{\boldsymbol{X}_k|\boldsymbol{Y}_{1:k},\boldsymbol{r}_{0:k}}(C\mid Y_{1:k},r_{0:k}^{(i)})$ are of the following form

$$\beta_i(C)=\begin{cases}\prod_{j=1}^{n_i}N(C;\underline{m}_j^{(i)},\mathrm{C}_j^{(i)}), & n_i\ge 1\\ 1, & n_i=0\end{cases}$$

and $N(\cdot;\cdot,\cdot)$ are Gaussian probability measures on $\mathbb{R}^d$.

Now, the set derivative of Eq. (27) can be written according to Eq. (23) as follows

$$\frac{\delta}{\delta\underline{x}}\left[\sum_{i=1}^N w^{(i)}\beta_i(\mathbb{R}^d)\right]=\sum_{i=1}^N w^{(i)}\frac{\delta\beta_i}{\delta x}(\mathbb{R}^d) \tag{28}$$

What remains to be shown is that

$$\frac{\delta\beta_i}{\delta\underline{x}}(\mathbb{R}^d)=\sum_{j=1}^{n_i}f_N(\underline{x};\underline{m}_j^{(i)},\mathrm{C}_j^{(i)}) \tag{29}$$

but this results directly from Eq. (26) and noticing that according to Eq. (25)

$$\frac{\delta}{\delta\underline{x}}P_i(\mathbb{R}^d)=P_i(\mathbb{R}^d)f_i(\underline{x})=f_i(\underline{x})$$

since $P_i(\mathbb{R}^d)=1$. Applying Eq. (29) back to Eq. (28) one obtains the result,

$$\frac{\delta}{\delta\underline{x}}\widehat{\beta}_{\boldsymbol{X}_k|\boldsymbol{Y}_{1:k}}(\mathbb{R}^d)=\sum_{i=1}^N w^{(i)}\sum_{j=1}^{n_i}f_N(\underline{x};\underline{m}_j^{(i)},\mathrm{C}_j^{(i)})$$

## References


[1] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, ser. B: Mathematical and Statistical Methods. AA Dordrecht, The Netherlands: Kluwer Academic Publishers, 1997, vol. 39.

[2] H. Sidenbladh and S.-L. Wirkander, "Tracking random sets of vehicles in terrain," in *Proceedings of the 2nd IEEE Workshop on Multi-Object Tracking*, Madison, WI, USA, 2003.

[3] ——, "Particle filtering for finite random sets," *IEEE Trans. Aerosp. Electron. Syst.*, 2004, (submitted manuscript).

[4] M. Vihola, "Random sets for multitarget tracking and data fusion," Licentiate thesis, Tampere University of Technology, Tampere, 2004. [Online]. Available: http://iki.fi/mvihola/publications.html

[5] ——, "Random set particle filter for bearings-only multitarget tracking," in *Signal Processing, Sensor Fusion, and Target Recognition XIV, Proceedings of SPIE*, I. Kadar, Ed., vol. 5809, Orlando, Florida, Mar. 2005, pp. 301–312.

[6] B.-N. Vo, S. Singh, and A. Doucet, "Random finite sets and sequential Monte Carlo methods in multi-target tracking," in *Proceedings of the International Conference on Radar*, Adelaide, Australia, Sept. 2003, pp. 486–491.

[7] ——, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.
[8] ——, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proceedings of the Sixth International Conference on Information Fusion*, vol. 2, Cairns, Australia, July 2003, pp. 792–799.
[9] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
[10] H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density," in *Proceedings of the Sixth International Conference on Information Fusion*, vol. 2, Cairns, Australia, July 2003, pp. 800–806.
[11] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. San Diego: Academic Press, 1988.
[12] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control*, Lake Louise, Alberta, Canada, Oct. 2000.
[13] L. D. Stone, C. A. Barlow, and T. L. Corwin, *Bayesian Multiple Target Tracking*. Norwood, MA: Artech House, 1999.
[14] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-Blackwellized particle filter for multiple target tracking," *Information Fusion*, 2006, (in press).
[15] ——, "Rao-Blackwellized Monte Carlo data association for multiple target tracking," in *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm, Sweden, June 2004, pp. 583–590.
[16] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.
[17] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs; CT: YBS Publishing, 1995.
[18] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers, 2000, pp. 176–183.
[19] R. Chen and J. S. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, pp. 493–508, 2000.
[20] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
[21] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Signal Processing Group, Department of Engineering, University of Cambridge, Cambridge, UK, Tech. Rep. CUED/F-INFENG/TR.310, 1998.
[22] K. P. Murphy, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.
[23] S. M. Herman, "A particle filtering approach to joint passive radar tracking and target classification," Ph.D. dissertation, University of Illinois, Urbana, Illinois, 2002.
[24] L. Råde and B. Westergren, *Beta Mathematics Handbook for Science and Engineering*, 4th ed. Lund, Sweden: Studentlitteratur, 1998.
[25] R. Mahler, *An Introduction to Multisource-Multitarget Statistics and its Applications*. Lockheed Martin, Mar. 2000.