The Rao-Blackwellized marginal M-SMC filter for Bayesian multi-target tracking and labelling

Edson Hiroshi Aoki*, Yvo Boers†, Lennart Svensson‡, Pranab K. Mandal* and Arunabha Bagchi*
*Department of Applied Mathematics, University of Twente, Enschede, The Netherlands
Email: {e.h.aoki, p.k.mandal, a.bagchi}@ewi.utwente.nl
†Thales Nederland B. V., Hengelo, The Netherlands
Email: yvo.boers@nl.thalesgroup.com
‡Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden
Email: lennart.svensson@chalmers.se

***Abstract*—In multi-target tracking (MTT), we are often interested not only in finding the position of the objects, but also allowing individual objects to be uniquely identified with the passage of time, by placing a label on each track. In some situations, however, observability conditions do not allow us to maintain the consistency in the correspondence between track labels and true objects.**

In this situation, it may be useful for the operator to know the probability of loss of this consistency, i.e. the probability of labelling error. This is theoretically possible using Bayesian multi-target tracking approaches like the Multi-target Sequential Monte Carlo (M-SMC) and the Multiple Hypothesis Tracking (MHT) filters, but unfortunately, it is well-known that these methods suffer from a form of degeneracy known as "self-resolving", that causes the probability of labelling error to be severely underestimated.

In this paper, we propose a new Sequential Monte Carlo algorithm for the multi-target tracking and labelling (MTTL) problem, the Rao-Blackwellized marginal M-SMC filter, that deals with self-resolving and is valid for multi-target scenarios with unknown/varying number of targets.

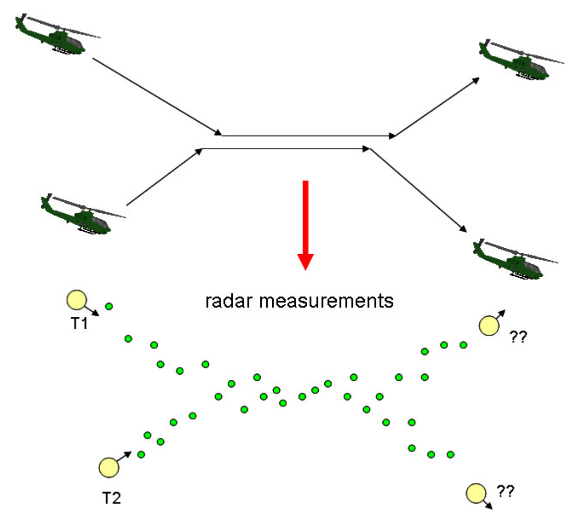


Fig. 1. Situation where assignment of labels to tracks is ambiguous

I. Introduction

The track labelling problem is perhaps just as old as the multi-target tracking problem itself. In the display of a radar operator, it is often necessary not only to display the estimated position of the multiple objects (i.e. the tracks), but also attribute a unique label to each track. Ideally, this track label should consistently be associated with the same real-world object, enhancing thus the situational awareness of the operator.

In practice, the feasibility of maintaining this label-to-true target consistency depends on observability conditions. One situation where this consistency is frequently lost is when targets move in close proximity to each other. In this case, the measurements and initial information may not allow us to precisely determine which target is which after the separation. Therefore, if required to make a hard decision to assign labels to tracks, the tracker will frequently make wrong choices. This situation (illustrated in Fig. 1), where the available information allows more than one labelling possibility is referred as "mixed labelling" by Boers, Sviestins and Driessen [1]. In that situation, two questions may be particularly relevant:

- **Question 1:** *How does one optimally assign labels T1 and T2 to the two tracks?*
- **Question 2:** *What is the probability that the assignment is incorrect, i.e. that track swap has occurred?* This probability may be useful to the operator; for instance, when a decision is only acceptable if we have high confidence that a target is who it seems to be.

This work tackles these two questions (together with the companion work [2]). The companion work focuses on conceptual aspects of the optimal labelling problem (including how to mathematically formulate **Questions 1 and 2**, as well as giving a clear physical interpretation to them), while this paper will propose a practical implementation. As one may expect, the companion paper will be repeatedly referred in this work.

In [2, Sect. 2], the Bayesian recursion of the MTTL problem is described using Finite Set Statistics (FISST). In principle, this recursion can be implemented using approximate Bayesian filters, such as the M-SMC filter [3]–[5] and the MHT [6]. In practice, as observed in previous works [1], [7], these algorithms suffer from the "self-resolving" phenomenon. This phenomenon causes the probability of incorrect labelling to be underestimated, such that a completely wrong answer may be given to **Question 2**.

For Bayesian estimation problems in general, "self-resolving" corresponds to the situation where uncertainties and ambiguities, that exist in the *exact posterior*, are for some reason not reflected in the output of the estimation algorithm. The phenomenon typically manifests in numerical filtering algorithms that approximate the posterior by a set of hypotheses or particles, and periodically prune them in order to avoid combinatorial explosion on the number of hypotheses/particles.

The phenomenon is known for some time in particle filter literature. Vermaak, Doucet and Pérez [8] have observed that a particle filter applied to a multi-modal distribution does not consistently maintain this multi-modality. Other problems that suffer from the same type of degeneration include parameter estimation [9], smoothing [10] and our specific problem of interest, track labelling [1].

In this work, we present a MTTL algorithm, the Rao-Blackwellized marginal M-SMC filter, that avoids self-resolving, and is applicable to general multi-target scenarios, i.e. with unknown and time-varying number of targets (although, as we are going to see, scenarios with target birth present additional challenges). Some MTTL algorithms capable of avoiding self-resolving have been proposed in recent works [7], [11], [12], but works [7], [12] are not consistent with the Bayesian formulation of the labelling problem that we derived in [2], and as we are going to mention, the algorithm presented in [11] is actually a special case of our proposed algorithm.

This paper is organized as follows. We finish this Introduction with a few notation conventions that are used throughout this work. Section II gives a conceptual introduction to the RBMPF, and Section III explains its adaptation to Random Finite Sets, namely the RBM M-SMC filter. Section IV describes the application of this technique to the track labelling problem, and discusses various practical aspects. Section V presents simulations, and Section VI draws conclusions.

Notation conventions: an upper-case letter (like X) denotes a vector-valued random variable, and its lower-case counterpart (like x) denotes a particular realization of X. An upper-case bold-faced letter (like $\mathbf{X}$) denotes a finite set-valued random variable, and its lower-case counterpart denotes the corresponding realization. The probability density of a vector-valued random variable X is denoted as $p(x)$; the multi-object density of a RFS variable (that we refer to simply as RFS density) is denoted as $f(\mathbf{x})$. We also use $E_z[g(X)]$ (or $E_z[g(\mathbf{X})]$) to denote the conditional expectation of a function $g(x)$ (or $g(\mathbf{x})$), conditioned on z.

II. Introduction to the Rao-Blackwellized Marginal Particle Filter

In this section, we provide an introduction to the Rao-Blackwellized marginal particle filter (RBMPF), which is a key element for our proposed solution to the MTTL problem. The RBMPF is a variant of the particle filter algorithm, designed to counter the self-resolving phenomenon. It has previously been applied [13] to the parameter estimation problem. The algorithm is essentially a combination of two well-known SMC methods: the Rao-Blackwellized particle filter (RBPF) [14] and the marginal particle filter[1] (MPF) [15].

Let X_k denote a state vector that is to be estimated, and Z^k the collection of measurements up to and including time k. Let us assume that the state X_k is composed of two parts S_k and L_k, i.e. $X_k = [S_k', L_k']'$. Observe that an expectation of a function $g(x_k)$ conditioned on Z^k is given by

$$\begin{aligned} E_{Z^k}[g(X_k)] &= \int\int g(s_k, l_k) p(l_k, s_k|Z^k) dl_k ds_k \\ &= \int\int g(s_k, l_k) p(l_k|s_k, Z^k) p(s_k|Z^k) dl_k ds_k. \end{aligned} \tag{1}$$

Now, let us assume that the decomposition $X_k = [S_k', L_k']'$ is such that we can solve the integral w.r.t. l_k, and that we approximate $p(s_k|Z^k)$ using the set of particles $\{w_k(i), s_k(i)\}_{i=1}^{N_P}$, where $w_k(i)$, $s_k(i)$ and N_P denote the particle weight, particle state and number of particles respectively. In such conditions, we approximate (1) as

$$E_{Z^k}[g(X_k)] \approx \sum_{i=1}^{N_P} w_k(i) \int g(s_k(i), l_k) p(l_k|s_k(i), Z^k) dl_k \tag{2}$$

i.e., we effectively approximate the posterior $p(x_k|Z^k)$ as

$$p(s_k, l_k|Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k(i) - s_k) p(l_k|s_k(i), Z^k). \tag{3}$$

The RBMPF is a Sequential Monte Carlo (SMC) method that iteratively obtains the approximation (3). How it is implemented depends on the particular characteristics of the Bayesian recursion of $p(x_k|Z^k)$, but a necessary condition is that the integral on l_k in (2) can be solved.

A justification of this approach, i.e. why it is effective against self-resolving, is presented in [13]. We present a justification more specific to our problem of interest (track labelling) in Section IV-G.

III. A Random Finite Set version of the RBMPF

To be able to apply the RBMPF technique to the MTTL problem, we need to come up with an approximation similar to (3), but for multi-object states instead. One convenient representation of such states is provided by Finite Set Statistics (FISST), which describes a scenario with multiple objects as a Random Finite Set (RFS) $\mathbf{X}_k = \left\{X_k^{(1)}, \ldots, X_k^{(T_k)}\right\}$, where T_k is the (random) number of objects and $X_k^{(i)}$, $i \in \{1, \ldots, T_k\}$, denotes the single-object state.

We will now derive a version of approximation (3) for RFS states, more specifically for the case where the single-object

[1] Not to be confused with the *marginalized* particle filter, which is another name for the Rao-Blackwellized particle filter.

state $X_k^{(i)}$ is hybrid continuous-discrete, i.e. it is given by $X_k^{(i)} = \left[S_k'^{(i)}, L_k'^{(i)}\right]'$, where L_k contains only components which assume values in a discrete state space. In order to do that, let us first consider the random finite sets $\mathbf{S}_k = \left\{S_k^{(1)}, \ldots, S_k^{(T_k)}\right\}$ and $\mathbf{L}_k = \{L_k^{(1)}, \ldots, L_k^{(T_k)}\}$, formed by the *partial states* (see [2, Def. 4.1]) of $\left\{X_k^{(1)}, \ldots, X_k^{(T_k)}\right\}$.

Definition 3.1: Let $\mathbf{s}_k = \left\{s_k^{(1)}, \ldots, s_k^{(t_k)}\right\}$ be a realization of $\mathbf{S}_k$, and $s_k = \left[s_k'^{(1)}, \ldots, s_k'^{(t_k)}\right]'$ be a vector formed by (arbitrarily) ordering the elements of $\mathbf{s}_k$. Similarly, let $l_k = \left[l_k'^{(1)}, \ldots, l_k'^{(t_k)}\right]'$ be a vector formed by ordering the elements of a realization $\mathbf{l}_k$ of $\mathbf{L}_k$. The $S^{(\cdot)}, L^{(\cdot)}$**-composition** of vectors s_k and l_k is defined as

$$h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k) \triangleq \left\{ \begin{bmatrix} s_k^{(1)} \\ l_k^{(1)} \end{bmatrix}, \ldots, \begin{bmatrix} s_k^{(t_k)} \\ l_k^{(t_k)} \end{bmatrix} \right\}, \qquad (4)$$

i.e. $h_{S^{(\cdot)},L^{(\cdot)}}$ is a special function that maps a pair of vectors to a finite set, more precisely to a realization of $\mathbf{X}_k$.

Proposition 3.2: Let $s_k = \left[s_k'^{(1)}, \ldots, s_k'^{(t_k)}\right]'$, and let us define the following collections:

$$\Omega_k(s_k) \triangleq \left\{l_k \middle| f\left(h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k)\middle|Z^k\right) > 0\right\} \qquad (5)$$

$$\Pi_k(s_k) \triangleq \left\{\mathbf{x}_k \middle| \exists l_k \in \Omega_k(s_k), \mathbf{x}_k = h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k)\right\} \qquad (6)$$

where $f(\cdot)$ denotes a RFS density (see Notation conventions in Section II).

Then, for any other vector s_k^* obtained by permuting the entries (of form $s_k^{(i)}$) of s_k, we have

$$\Pi_k(s_k) = \Pi_k(s_k^*). \qquad (7)$$

Proof: Let us suppose that we obtained s_k^* by applying a permutation map $m : \{1, \ldots, t_k\} \to \{1, \ldots, t_k\}$ to s_k. Now, observe that if we obtain l_k^* from some l_k using the same map m, we will have

$$h_{S^{(\cdot)},L^{(\cdot)}}(s_k^*, l_k^*) = h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k). \qquad (8)$$

Since (8) holds for every $l_k \in \Omega_k(s_k)$ (and thus for every element of $\Pi_k(s_k)$), and $h_{S^{(\cdot)},L^{(\cdot)}}(s_k^*, l_k^*) \in \Pi_k(s_k^*)$, we can conclude that $\Pi_k(s_k) \subset \Pi_k(s_k^*)$. $\Pi_k(s_k^*) \subset \Pi_k(s_k)$ follows similarly. ■

Definition 3.3: For a given $\mathbf{s}_k$, let us define the collection

$$\Pi_k(\mathbf{s}_k) \triangleq \Pi_k(s_k) \qquad (9)$$

where s_k is a vector formed by (arbitrarily) ordering the elements of $\mathbf{s}_k$ and $\Pi_k(s_k)$ is given by (6). The fact that $\Pi_k(\mathbf{s}_k)$ is well-defined comes from Proposition 3.2.

Theorem 3.4: A conditional expectation of form $E_{Z^k}[g(\mathbf{X}_k)]$, where g denotes a set function, can be written as

$$\begin{aligned} &E_{Z^k}[g(\mathbf{X}_k)] \\ &= \int \sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k)} g(\mathbf{x}_k) f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k) f(\mathbf{s}_k|Z^k) \delta\mathbf{s}_k \end{aligned} \qquad (10)$$

where $f_{L^{(\cdot)}|S^{(\cdot)}}$ denotes a $L^{(\cdot)}|S^{(\cdot)}$-split density (defined in [2, Def. 4.2]).

Proof: First, observe that, using the definition of set integral [5, pp. 361–362]:

$$\begin{aligned} &E_{Z^k}[g(\mathbf{X}_k)] \\ &= \int g(\mathbf{x}_k) f(\mathbf{x}_k|Z^k) \delta\mathbf{x}_k \\ &= \sum_{t_k=0}^{\infty} \frac{1}{t_k!} \int \sum_{l_k \in \Omega_k(s_k)} g(h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k)) \\ &\quad \times f(h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k)|Z^k) ds_k \\ &= \sum_{t_k=0}^{\infty} \frac{1}{t_k!} \int \sum_{\mathbf{x}_k \in \Pi_k(s_k)} g(\mathbf{x}_k) f(\mathbf{x}_k|Z^k) ds_k. \end{aligned} \qquad (11)$$

From [2, Def. 4.2]:

$$f(\mathbf{x}_k|Z^k) = f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k) f(\mathbf{s}_k|Z^k) \qquad (12)$$

where for $\mathbf{x}_k = h_{S^{(\cdot)},L^{(\cdot)}}(s_k, l_k)$, $\mathbf{s}_k$ is a finite set formed by the entries of vector s_k. It follows that

$$\begin{aligned} &E_{Z^k}[g(\mathbf{X}_k)] \\ &= \sum_{t_k=0}^{\infty} \frac{1}{t_k!} \int \sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k)} g(\mathbf{x}_k) f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k) f(\mathbf{s}_k|Z^k) ds_k \\ &= \int \sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k)} g(\mathbf{x}_k) f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k) f(\mathbf{s}_k|Z^k) \delta\mathbf{s}_k. \end{aligned} \qquad (13)$$

■

Theorem 3.4 allows us to obtain an approximation similar to (3) for RFS densities, for the case of hybrid continuous-discrete single-object states. If we approximate $f(\mathbf{s}_k|Z^k)$ using the set of particles $\{w_k(i), \mathbf{s}_k(i)\}_{i=1}^{N_P}$, we may approximate the conditional expectation $E_{Z^k}[g(\mathbf{X}_k)]$ as

$$E_{Z^k}[g(\mathbf{X}_k)] \approx \sum_{i=1}^{N_P} w_k(i) \sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))} g(\mathbf{x}_k) f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k) \qquad (14)$$

and the multi-target posterior $f(\mathbf{x}_k|Z^k)$ is therefore approximated as

$$f(\mathbf{x}_k|Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(\mathbf{s}_k(i) - \mathbf{s}_k) f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k) \qquad (15)$$

with the elements of $\mathbf{s}_k$ assumed to be partial states of distinct elements of $\mathbf{x}_k$.

Since the RFS version of the "plain-vanilla" particle filter is referred as Multi-target Sequential Monte Carlo (M-SMC) filter [5, pp. 551–564], we are going to refer to the algorithm that iteratively obtains (15) as Rao-Blackwellized Marginal M-SMC (RBM M-SMC) filter.

IV. The RBM M-SMC filter applied to the track labelling problem

Now that we have conceptually introduced the RBM M-SMC filter, we are ready to present its implementation to the MTTL problem. We will begin the Section by briefly describing the FISST formulation of the MTTL problem (for a full discussion, see [2, Sec. 2]). We will then explain the method (Sections IV-B to IV-F), and thereafter describe its theoretical justification (Section IV-G) and performance aspects (Section IV-H).

A. Review of MTTL problem formulation

For the RFS $\mathbf{X}_k$, let the single-target state be given by $X_k^{(i)} = \left[S_k^{\prime(i)}, L_k^{\prime(i)}\right]^{\prime}$, where $L_k^{(i)}$ denotes the assigned label to the target, and $S_k^{(i)}$ denotes all other state components (position, velocity, etc.). With appropriate Markov assumptions, the Bayesian recursion for the RFS density $f(\mathbf{x}_k|Z^k)$ has the form

$$f(\mathbf{x}_k|Z^k) = \frac{f(\mathbf{z}_k|\mathbf{x}_k)f(\mathbf{x}_k|Z^{k-1})}{f(\mathbf{z}_k|Z^{k-1})} \tag{16}$$

where $\mathbf{z}_k$ denotes the most recent set of observations, $f(\mathbf{z}_k|\mathbf{x}_k)$ is the multi-object likelihood function and

$$f(\mathbf{x}_k|Z^{k-1}) = \int f(\mathbf{x}_k|\mathbf{x}_{k-1})f(\mathbf{x}_{k-1}|Z^{k-1})\delta\mathbf{x}_{k-1} \tag{17}$$

$$f(\mathbf{z}_k|Z^{k-1}) = \int f(\mathbf{z}_k|\mathbf{x}_k)f(\mathbf{x}_k|Z^{k-1})\delta\mathbf{x}_k. \tag{18}$$

We also make two additional assumptions:

$$f(\mathbf{z}_k|\mathbf{x}_k) = f(\mathbf{z}_k|\mathbf{s}_k), \tag{19}$$

$$f(\mathbf{s}_k|\mathbf{x}_{k-1}) = f(\mathbf{s}_k|\mathbf{s}_{k-1}). \tag{20}$$

where, in (19), the elements of $\mathbf{s}_k$ are partial states of distinct elements of $\mathbf{x}_k$, and both finite sets have the same cardinality. The same property holds for $\mathbf{s}_{k-1}$ and $\mathbf{x}_{k-1}$ in (20). These assumptions are not restrictive: see [16, Sec. 2.1, 2.3].

B. The RBM M-SMC approximation applied to the MTTL problem

We remark that $f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|Z^k)$ corresponds to the *labelling probability* (defined in [2, Def. 4.4]), that we write as $p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k)$, with all elements of $\mathbf{s}_k$ assumed to be partial states of distinct elements of $\mathbf{x}_k$.

From (14), the expectation of a set function $g(\mathbf{x}_k)$ conditioned on Z^k can then be approximated as

$$E_{Z^k}[g(\mathbf{X}_k)] \approx \sum_{i=1}^{N_P} w_k(i) \sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))} g(\mathbf{x}_k) p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k(i)) \tag{21}$$

and the set of particles produced by the algorithm at each time step k is given by

$$\left\{\mathbf{s}_k(i), w_k(i), \{p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k(i))\}_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))}\right\}_{i=1}^{N_P}. \tag{22}$$

As mentioned in [2, Remark 4.3], $p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k)$ corresponds to a conditional probability mass, such that we have

$$\sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))} p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k(i)) = 1. \tag{23}$$

We will now describe how the components of the set of particles (22) are calculated.

C. Calculation of particle states and weights

In order to obtain $\{\mathbf{s}_k(i), w_k(i)\}_{i=1}^{N_P}$, we need to characterize the Bayesian recursion for $f(\mathbf{s}_k|Z^k)$. We are going to use the following result (see detailed proof in [16, Sect. 2.3]): for $\mathbf{X}_k$, $\mathbf{Z}_k$ and $\mathbf{S}_k$ according to the problem formulation of Section IV-A, the time series $\{(\mathbf{S}_k, \mathbf{Z}_k)\}$ consists of a first-order partially observed Markov process, i.e.

$$f(\mathbf{s}_k|Z^k) = \frac{f(\mathbf{z}_k|\mathbf{s}_k)f(\mathbf{s}_k|Z^{k-1})}{f(\mathbf{z}_k|Z^{k-1})} \tag{24}$$

where

$$f(\mathbf{s}_k|Z^{k-1}) = \int f(\mathbf{s}_k|\mathbf{s}_{k-1})f(\mathbf{s}_{k-1}|Z^{k-1})\delta\mathbf{s}_{k-1}. \tag{25}$$

Therefore, in order to implement recursion (24) using a marginal particle filter, we need to specify $f(\mathbf{z}_k|\mathbf{s}_k)$, $f(\mathbf{s}_0)$ and $f(\mathbf{s}_k|\mathbf{s}_{k-1})$. Formulas for these densities for various multi-target models can be found in [5, chap. 12, 13].

D. Calculation of particle labelling probabilities

First, observe that, if $\mathbf{x}_k \in \Pi_k(\mathbf{s}_k)(i)$

$$p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k(i)) = \frac{f(\mathbf{x}_k|Z^k)}{f(\mathbf{s}_k(i)|Z^k)} \tag{26}$$

and through a few manipulations, it is possible to show that

$$p_{\text{l}}(\mathbf{x}_k|\mathbf{s}_k(i)) = \frac{f(\mathbf{x}_k|Z^{k-1})}{f(\mathbf{s}_k(i)|Z^{k-1})}. \tag{27}$$

Note that the denominator of (27) is constant for a given particle i, and due to property (23), it can be taken into account by normalizing the labelling probabilities for each particle. We therefore only need to look at the numerator of (27). We may expand it as

$$f(\mathbf{x}_k|Z^{k-1}) = \int f(\mathbf{x}_k|\mathbf{x}_{k-1})f(\mathbf{x}_{k-1}|Z^{k-1})\delta\mathbf{x}_{k-1} \tag{28}$$

and if we assume that $f(\mathbf{x}_{k-1}|Z^{k-1})$ is approximated by the set of particles

$$\Big\{\mathbf{s}_{k-1}(j), w_{k-1}(j), \{p_{\text{l}}(\mathbf{x}_{k-1}|\mathbf{s}_{k-1}(j))\}_{\mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j))}\Big\}_{j=1}^{N_P} \tag{29}$$

we can apply (21) to approximate (28) as

$$f(\mathbf{x}_k|Z^{k-1}) \approx \sum_{j=1}^{N_P} w_{k-1}(j) \times \sum_{\mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j))} f(\mathbf{x}_k|\mathbf{x}_{k-1}) p_{\text{l}}(\mathbf{x}_{k-1}|\mathbf{s}_{k-1}(j)). \tag{30}$$

Note that the labelled multi-target state transition density $f(\mathbf{x}_k|\mathbf{x}_{k-1})$ can have a quite different form from its unlabelled counterpart $f(\mathbf{s}_k|\mathbf{s}_{k-1})$. Formulas for $f(\mathbf{x}_k|\mathbf{x}_{k-1})$ for some cases of interest can be found in [2, Sect. 2.2].

E. Algorithm

Initialization: For each particle $i = 1, \ldots, N_P$

1) Sample $\mathbf{s}_0(i) \sim f(\mathbf{s}_0)$
2) Make $w_0(i) = \frac{1}{N_P}$
3) For each $\mathbf{x}_0 \in \Pi_0(\mathbf{s}_0(i))$, set $p_{\mathrm{l}}(\mathbf{x}_0|\mathbf{s}_0(i))$.

At every time step k**:**

1) For each particle $i = 1, \ldots, N_P$
   a) Sample $\mathbf{s}_k(i) \sim \sum_{j=1}^{N_P} w_{k-1}(j) q(\mathbf{s}_k|\mathbf{s}_{k-1}(j), \mathbf{z}_k)$, where $q(\mathbf{s}_k|\mathbf{s}_{k-1}, \mathbf{z}_k)$ is the MPF importance sampling function
   b) Calculate the unnormalized weight according to

$$\overline{w}_k(i) = \frac{f(\mathbf{z}_k|\mathbf{s}_k(i)) \sum_{j=1}^{N_P} w_{k-1}(j) f(\mathbf{s}_k(i)|\mathbf{s}_{k-1}(j))}{\sum_{j=1}^{N_P} w_{k-1}(j) q(\mathbf{s}_k|\mathbf{s}_{k-1}(j), \mathbf{z}_k)} \tag{31}$$

   (refer to [5, chap. 12, 13] for formulas for $f(\mathbf{z}_k|\mathbf{s}_k)$ and $f(\mathbf{s}_k|\mathbf{s}_{k-1})$)
   c) For each $\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))$, compute the unnormalized particle labelling probability according to

$$\overline{p}_{\mathrm{l}}(\mathbf{x}_k|\mathbf{s}_k(i)) = \sum_{j=1}^{N_P} w_{k-1}(j) \times \sum_{\mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j))} f(\mathbf{x}_k|\mathbf{x}_{k-1}) p_{\mathrm{l}}(\mathbf{x}_{k-1}|\mathbf{s}_{k-1}(j)) \tag{32}$$

   (refer to [2, Sect. 2.2] for formulas for $f(\mathbf{x}_k|\mathbf{x}_{k-1})$)
   d) Normalize the particle labelling probabilities according to

$$p_{\mathrm{l}}(\mathbf{x}_k|\mathbf{s}_k(i)) = \frac{\overline{p}_{\mathrm{l}}(\mathbf{x}_k|\mathbf{s}_k(i))}{\sum_{\tilde{\mathbf{x}}_k \in \Pi_k(\mathbf{s}_k(i))} \overline{p}_{\mathrm{l}}(\tilde{\mathbf{x}}_k|\mathbf{s}_k(i))} \tag{33}$$

2) Normalize the particle weights according to

$$w_k(i) = \frac{\overline{w}_k(i)}{\sum_{j=1}^{N_P} \overline{w}_k(j)} \tag{34}$$

Interestingly, it can be shown that the auxiliary variable marginal PF with mirror particles from García-Fernández, Morelande and Grajal [11] is a special case of the RBM M-SMC filter, specifically the case where the number of targets is known and equal to two.

F. Track extraction

Like in all SMC methods, the raw output of the RBM M-SMC filter is an approximation of the posterior density. An additional step is required to obtain the output to be displayed to the user, in our case, the set of labelled tracks.

In [2, Sec. 4.2], a conceptual method for track extraction in general MTTL algorithms, the MMOSPA-MLP (Minimum Mean Optimal Subpattern Assignment - Maximum Labelling Probability), has been proposed. As described in the same work, this method has certain advantages: it avoids the track coalescence problem, can be used in general scenarios with unknown and/or time-varying number of targets, and its results have clear physical interpretation. We will now describe how to implement the MMOSPA-MLP estimate for the RBM M-SMC filter.

In order to implement the MMOSPA step, we evaluate, for each particle i, the value of the MOSPA function, which can be approximated according to

$$\begin{aligned} \text{MOSPA}(i) &= \int \left(\epsilon_p^{(c)}(\mathbf{s}_k, \mathbf{s}_k(i))\right)^p f(\mathbf{s}_k|Z^k)\delta\mathbf{s}_k \\ &\approx \sum_{j=1}^{N_P} w_k(j) \left(\epsilon_p^{(c)}(\mathbf{s}_k(j), \mathbf{s}_k(i))\right)^p \end{aligned} \tag{35}$$

where $\epsilon_p^{(c)}$ is the Optimal Subpattern Assignment (OSPA) metric defined by Schuhmacher, Vo and Vo [17] and c and p are parameters discussed in the same work. We then select the MMOSPA estimate (consisting of a set of unlabelled tracks) according to:

$$\hat{\mathbf{s}}_k = \arg\min_{\mathbf{s}_k(i)} \text{MOSPA}(i) \tag{36}$$

and finally, the MMOSPA-MLP estimate (consisting of a set of labelled tracks) is given by

$$\hat{\mathbf{x}}_k = \arg \max_{\mathbf{x}_k \in \Pi_k(\hat{\mathbf{s}}_k)} p_{\mathrm{l}}(\mathbf{x}_k|\hat{\mathbf{s}}_k). \tag{37}$$

Observe that, from the physical interpretation of labelling probabilities (see [2, Remark 4.5]) the labelling probability $p_{\mathrm{l}}(\hat{\mathbf{x}}_k|\hat{\mathbf{s}}_k)$ of an estimate $\hat{\mathbf{x}}_k$ is the probability of the assignment of labels to states in $\hat{\mathbf{x}}_k$, under the assumption that the unlabelled states $\hat{\mathbf{s}}_k$ match the true target states.

G. Theoretical justification

To understand the rationale behind using the RBM M-SMC filter for the MTTL problem, let us recall that our goal is to prevent the self-resolving phenomenon of particle filters from obscuring the mixed labelling contained in the multi-target posterior. In other words, if there is more than one relevant possibility on labels $l_k^{(1)}, \ldots, l_k^{(t_k)}$ being assigned to the unlabelled tracks $\hat{s}_k^{(1)}, \ldots, \hat{s}_k^{(t_k)}$, we need to prevent these labelling hypotheses from being eliminated by the particle filter mechanism.

The Rao-Blackwellized particle filter is an intuitive way to accomplish that. By making the particle approximation apply only to $f(\mathbf{s}_k|Z^k)$, and analytically keeping track of all labelling probabilities for each multi-target unlabelled state hypothesis $\mathbf{s}_k(i)$, we effectively prevent labelling possibilities from disappearing during the resampling process. A more tricky question is why the RB M-SMC filter also needs to be a *marginal* particle filter.

The answer is that a RB M-SMC filter ("non-marginal") will *unavoidably* result in degenerate estimates. This is because with this algorithm, is possible to show that the particle approximation of the conditional expectation of a function $g(\mathbf{x}_k)$, instead of being given by (14), would be given by

$$\begin{aligned} E_{Z^k}[g(\mathbf{X}_k)] \approx \sum_{i=1}^{N_P} w_k(i) &\sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))} g(\mathbf{x}_k) \\ &\times f_{L^{(\cdot)}|S^{(\cdot)}}(\mathbf{x}_k|\mathbf{s}_0(i), \ldots, \mathbf{s}_{k-1}(i), Z^k). \end{aligned} \tag{38}$$

The problem of approximation (38) is that the last term is conditioned on the past of unlabelled state trajectories $\mathbf{s}_0(i), \ldots, \mathbf{s}_{k-1}(i)$. But the resampling mechanism of the particle filter inherently biases the statistical information about past states, resulting in degeneracy of the estimate $E_{Z^k}[g(\mathbf{X}_k)]$. The use of a RBM M-SMC filter eliminates the explicit dependency on the past trajectories $\mathbf{s}_0(i), \ldots, \mathbf{s}_{k-1}(i)$, sufficing, in principle, that the approximation of $f(\mathbf{s}_k|Z^k)$ is good enough.

Remark 4.1: The obvious assumption here is that the particle approximation of $f(\mathbf{s}_k|Z^k)$ is good, which requires good mixing properties of the system $(\mathbf{S}_k, \mathbf{Z}_k)$. This may not be the case, for instance, if the single-object state $S_k^{(i)}$ contains static or slowly varying components (for instance, target classification).

H. Computational and practical aspects

The RBM M-SMC filter has formidable computational cost. It is suited to deal with labelling issues in small scale scenarios, such as tracking a small group of targets, where the targets may approach and separate from each other. It is certainly not suited for large scale scenarios.

The biggest computational burden of the algorithm is computing the particle labelling probabilities $p_{\mathrm{l}}(\mathbf{x}_k|\mathbf{s}_k(i))$. From (32), the complexity of calculating a single labelling probability is about $O(N_P N_\Pi)$ where N_Π denotes the typical cardinality of the sets $\Pi_k(\cdot)$. Hence, to compute all labelling probabilities from all particles, the complexity would be about $O(N_P^2 N_\Pi^2)$.

How large N_Π could be? If there are no target births and deaths, and the number of targets known and equal to t, then $N_\Pi = t!$. Therefore, starting from 10 targets, we will already have millions of labelling hypotheses.

Remark 4.2: Like the standard M-SMC filter, the RBM M-SMC filter does not impose any restriction on the properties of the multi-target state transition model $f(\mathbf{x}|\mathbf{x}_{k-1})$. However, we remark that from the conclusions of [2], the problem of target birth with labelling still needs to be better understood from a theoretical point of view. Therefore, it is too early to jump into conclusions about the suitability of the algorithm to deal with scenarios with target birth, although it has no apparent problems to deal with target death.

V. Simulations

Empirical studies on the tracking problem are typically carried on by performing multiple Monte Carlo runs (with different sequences of measurements) and comparing the tracking results with the ground truth. In our case, however, we want to analyze not the ability of the algorithms of matching the ground truth, but instead their ability of identifying the uncertainty in label-to-track association that exists in the exact (i.e. Bayes-calculated) multi-target posterior $f(\mathbf{x}_k|Z^k)$.

For the case of two targets, and hence with the filter output given by the pair of tracks $\hat{\mathbf{x}}_k = \left\{\hat{x}_k^{(1)}, \hat{x}_k^{(2)}\right\}$, this uncertainty can be described by the probability of track swap, given by $1 - p_{\mathrm{l}}(\hat{\mathbf{x}}_k|\hat{\mathbf{s}}_k)$, where $\hat{\mathbf{s}}_k$ denotes the corresponding pair of unlabelled tracks.

A. Dealing with the unknown ground truth

Ideally, we could evaluate an algorithm by testing it against a sequence of measurements Z^k, and thereafter comparing the labelling probability $p_{\mathrm{l}}(\hat{\mathbf{x}}_k|\hat{\mathbf{s}}_k)$ calculated by the algorithm with its "true" value, which should come from $f(\mathbf{x}_k|Z^k)$. Unfortunately, we do not know the true value of $p_{\mathrm{l}}(\hat{\mathbf{x}}_k|\hat{\mathbf{s}}_k)$ since we do not have an exact Bayes estimator. To deal with the missing ground truth, we can use different sorts of evidence, including:

1) *Evidence from the exact Bayes recursion:* For the situation where targets share exactly the same state for some time, plus some additional assumptions on the scenario, we know, from the analysis on the exact multi-target Bayes recursion in [18, Sect. III.B], that we must have "total mixed labelling" (as defined in [2, Sect. 3.1]). In the same situation, we also know from [18, Lemma 3.5] that unless some special conditions are met, mixed labelling will never disappear. Therefore, we know that a filter that computes a track labelling probability that represent this "total mixed labelling" situation (for instance, 50% track swap probability for the two-target case) is yielding correct results;
2) *Evidence from simulations with larger number of particles:* Another way to assess correctness of a proposal multi-target filter is to run simulations using the SIR M-SMC filter with increasingly larger number of particles, and check whether the results appear to converge (or not) to the results of the proposal multi-target filter.

B. Simulation description

The following scenarios, shown in Fig. 2, are analyzed:

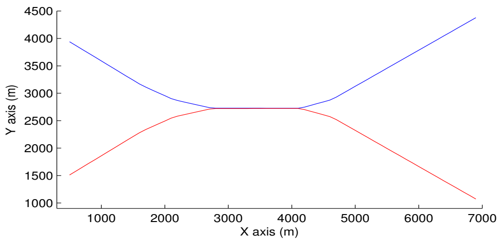
1) Two targets that approach each other, become exactly adjacent, and then separate;
2) Two targets that approach each other, keep somewhat apart, and then separate;

For both scenarios, we consider that the number of targets is known by the tracker. The single-target unlabelled state has form $S^{(i)} = \left[P_{\mathrm{x}}^{(i)}, P_{\mathrm{y}}^{(i)}, V_{\mathrm{x}}^{(i)}, V_{\mathrm{y}}^{(i)}\right]'$, where x and y denote the Cartesian coordinates, and $\left(P_{\mathrm{x}}^{(i)}, P_{\mathrm{y}}^{(i)}\right)$ and $\left(V_{\mathrm{x}}^{(i)}, V_{\mathrm{y}}^{(i)}\right)$ correspond respectively to the position and velocity components.

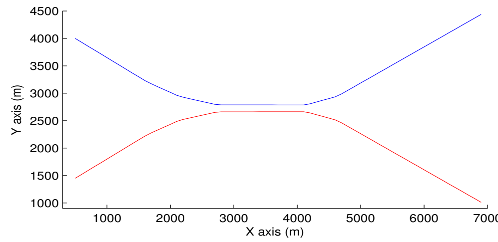
The multi-target measurement model $f(\mathbf{y}_k|\mathbf{s}_k)$ corresponds to the detection-type measurement model with no missed detections or false alarms (described in [5, Sect. 12.3.4]). The observation period is 2 seconds, and the single-measurement, single-target likelihood function is given by

$$p\left(z_k^{(i)} \middle| s_k^{(j)}\right) = \mathcal{N}\left(z_k^{(i)}; \begin{bmatrix} p_{\mathrm{x}}^{(j)} \\ p_{\mathrm{y}}^{(j)} \end{bmatrix}, \begin{bmatrix} 676 & 0 \\ 0 & 676 \end{bmatrix}\right). \quad (39)$$

The single-target Markov model corresponds to the discretized White Noise acceleration model described in [19], with a power spectral density of 676. For all scenarios, we simulate both the SIR M-SMC and the RBM M-SMC filter. For both algorithms, we use 2,000 particles, initiated near

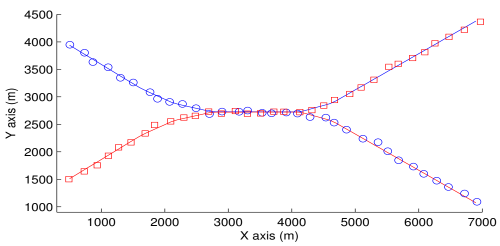


(a) Scenario 1

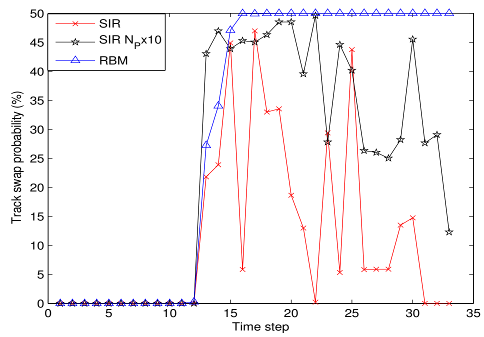


(b) Scenario 2

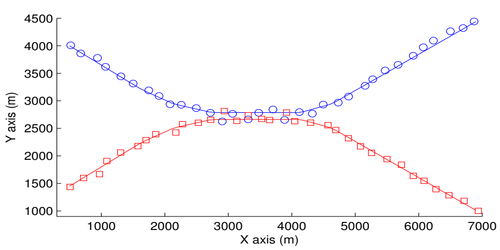
Fig. 2. Multi-target simulation scenarios



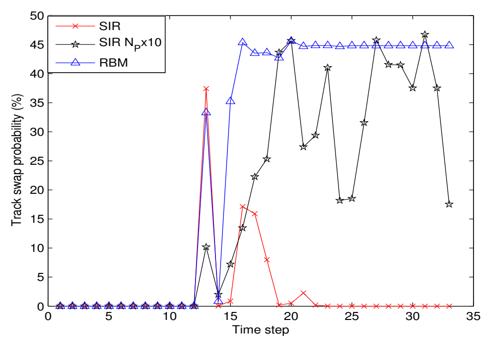
(a) Scenario 1: MMOSPA-MLP using RBM



(b) Scenario 1: Estimated track swap probabilities



(c) Scenario 2: MMOSPA-MLP using RBM



(d) Scenario 2: Estimated track swap probabilities

Fig. 3. Simulation results

the true locations of targets. Moreover, we will simulate an additional run of the SIR M-SMC filter using 10 times more particles, that we will henceforth refer as "SIR $N_P \times 10$".

For both particle filters, we use the Markov density as importance sampling function, i.e. $f(\mathbf{x}_k(i)|\mathbf{x}_{k-1}(i))$ for the SIR M-SMC filter, and $f(\mathbf{s}_k(i)|\mathbf{s}_{k-1}(j))$ for the RBM M-SMC filter.

C. Simulation results for each scenario

1) Scenario 1: Fig. 3(a) shows the MMOSPA-MLP estimate calculated by the RBM M-SMC filter. Although this run results in a track swap, we remark that the figure is only included here for illustrative purposes; since this scenario represents a "total confusion" situation, we know, from evidence 1 of Section V-A, that the "true" probability of track swap is around 50%; therefore, whether an algorithm produces or not the correct assignment of labels for a single run is statistically irrelevant. In the figure, the continuous lines correspond to the true trajectories of targets, and the circles/squares denote the MMOSPA-MLP tracks. For these tracks, each different color/symbol combination corresponds to a different assigned track label.

To compare the three algorithms, we will look instead at Fig. 3(b) that shows the track swap probabilities for the MMOSPA-MLP estimates calculated by the three algorithms. Although these probabilities are rigorously not comparable (since they are not based on the same values of $\hat{\mathbf{x}}_k$), nonetheless they provide valuable information about the behavior of each filter.

Using evidence 1 of Section V-A, we can see that only the RBM M-SMC results in the correct probability of track swap of 50%. Clearly, both SIR M-SMC filters are affected by the self-resolving phenomenon, with the ambiguity in label-to-track association being severely underestimated. Note that, as predicted from evidence 2, the SIR $N_P \times 10$ leads to better results: its computed track swap probabilities are significatly higher than the SIR. It is also easy to see, from Fig. 3(b), that using a SIR with more particles only postpones, but does not prevent, the self-resolving phenomenon.

2) Scenario 2: Note that, since in this scenario there seems to be only partial confusion of target statess, we cannot assess correctness purely on basis of evidence 1. Let us then give a close look at Fig. 3(d). We can see that, by increasing the number of particles of the SIR M-SMC filter, the calculated track swap probabilities become closer to the result of the RBM M-SMC filter. Since from evidence 2, increasing the number of particles of a SIR M-SMC filter should lead to better accuracy, we can at least say that the RBM M-SMC filter leads to more accurate track swap probabilities than the SIR for the same number of particles.

Besides, the RBM M-SMC filter, unlike the SIR M-SMC filers, maintains the same track swap probability after the targets separate. This behavior seems appropriate, since measurements subsequent to target separation would not be informative w.r.t. the true target identities. Hence, we can assess that the decline of the track swap probability observed in the two SIR M-SMC filters is due to self-resolving.

VI. Conclusion

In this paper over the MTTL problem, we followed our theoretical discussion in [2] with the proposition of a novel Sequential Monte Carlo solution for this problem. In order to design this solution, we derived an extension of the Rao-Blackwellized Marginal particle filter, that, we believe, may also be useful for different applications. The experimental results show that the proposed algorithm, the RBM M-SMC filter, is indeed far more suitable to answering the questions that we proposed in Section I (and were mathematically formulated in [2]), than the "plain vanilla" particle filter implementation of the MTTL problem, i.e. the SIR M-SMC filter.

An interesting future work would be to evaluate (or adapt) the algorithm for scenarios with target birth, but before doing that, we plan first to give a better theoretical look at the problem of target birth with labelling, as we mentioned in [2]. Another possible future work is to find adequate performance measures for evaluating scenarios with only "partial mixed labelling", such that we can more precisely assess the accuracy of labelling probabilities for such scenarios.

Acknowledgments

The research leading to these results has received funding from the EU's Seventh Framework Programme under grant agreement n° 238710. The research has been carried out in the MC IMPULSE project: https://mcimpulse.isy.liu.se.

This research has been also supported by the Netherlands Organisation for Scientific Research (NWO) under the Casimir program, contract 018.003.004. Under this grant Yvo Boers holds a part-time position at the Department of Applied Mathematics at the University of Twente.

We also thank Hans Driessen (Thales Nederland B.V.) for the valuable contributions.

References

[1] Y. Boers, E. Sviestins, and J. N. Driessen, "Mixed labelling in multitarget particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 2, pp. 792–802, 2010.

[2] E. H. Aoki, Y. Boers, L. Svensson, P. K. Mandal, and A. Bagchi, "A Bayesian look at the optimal track labelling problem," in *Proc. DF&TT'12*, London, UK, May 16–17, 2012.

[3] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, 2005.

[4] M. Vihola, "Rao-Blackwellised particle filtering in random set multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 689–705, 2007.

[5] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Noorwood, MA: Artech House, 2007.

[6] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, Dec. 1979.

[7] D. Crouse, P. Willett, L. Svensson, D. Svensson, and M. Guerriero, "The set MHT," in *Proc. FUSION 2011*, Chicago, IL, Jul. 5–8, 2011.

[8] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining multi-modality through mixture tracking," in *Proc. 9th IEEE International Conference on Computer Vision*, vol. 2, Nice, France, 2003, pp. 1110–1116.

[9] H. A. P. Blom and E. A. Bloem, "Particle filtering for stochastic hybrid systems," in *Proc. 43rd IEEE Conf. Decision and Control*, Atlantis, Bahamas, Dec. 14–17, 2004.

[10] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space model," *Annals Institute Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010.

[11] A. García-Fernández, M. Morelande, and J. Grajal, "Particle filter for extracting target label information when targets move in close proximity," in *Proc. FUSION 2011*, Chicago, IL, Jul. 5–8, 2011.

[12] H. A. P. Blom and E. A. Bloem, "Decomposed particle filtering and track swap estimation in tracking two closely spaced targets," in *Proc. FUSION 2011*, Chicago, IL, Jul. 5–8, 2011.

[13] F. Lindsten, T. B. Schön, and L. Svensson, "A non-degenerate Rao-Blackwellised particle filter for estimating static parameters in dynamical models," in *Proc. 16th IFAC Symposium on System Identification (SYSID)*, 2012.

[14] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *J. Royal Stat. Soc. B*, vol. 64, pp. 827–836, 2002.

[15] M. Klaas, N. de Freitas, and A. Doucet, "Toward practical N^2 Monte Carlo: the marginal particle filter," in *Proc. 21th Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. Arlington, Virginia: AUAI Press, 2005, pp. 308–315.

[16] E. H. Aoki, Y. Boers, L. Svensson, P. K. Mandal, and A. Bagchi, "An analysis of the Bayesian track labelling problem," University of Twente, Enschede, The Netherlands, Tech. Rep. 1980, 2012. [Online]. Available: http://www.math.utwente.nl/publications

[17] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, 2008.

[18] E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers, "A theoretical analysis of Bayes-optimal multi-target tracking and labelling," University of Twente, Enschede, The Netherlands, Tech. Rep. 1953, 2011. [Online]. Available: http://www.math.utwente.nl/publications

[19] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York, NY: John Wiley & Sons, 2001, ch. 6.