Random Set Particle Filter for Bearings-only Multitarget Tracking

Matti Vihola

Datactica Ltd, Hermiankatu 1, 33720 Tampere, Finland

ABSTRACT

The random set approach to multitarget tracking is a theoretically sound framework that covers joint estimation of the number of targets and the state of the targets. This paper describes a particle filter implementation of the random set multitarget filter. The contribution of this paper to the random set tracking framework is the formulation of a measurement model where each sensor report is assumed to contain at most one measurement. The implemented filter was tested in synthetic bearings-only tracking scenarios containing up to two targets in the presence of false alarms and missed measurements. The estimated target state consisted of 2D position and velocity components. The filter was capable to track the targets fairly well despite of the missing measurements and the relatively high false alarm rates. In addition, the filter showed robustness against wrong parameter values of false alarm rates. The results that were obtained during the limited tests of the filter show that the random set framework has potential for challenging tracking situations. On the other hand, the computational burden of the described implementation is quite high and increases approximately linearly with respect to the expected number of targets.

Keywords: Bearings-only tracking, finite set statistics, multitarget tracking, particle filter

1. INTRODUCTION

Many modern multitarget tracking algorithms are based on more or less separate association, filtering and track initiation algorithms. Challenging tracking situations may be hard to tackle by the conventional heuristics-based track initiation methods. The theory of finite random sets provides a solid mathematical basis for the multitarget estimation.[1] In brief, the random set approach formulates the whole multitarget tracking problem as one Bayesian estimation problem. The random set approach is, as many top-down strategies are, computationally challenging. Recently, there has been several publications that describe sequential Monte Carlo approximations of the random set filtering problem.[2, 3]

This paper introduces a sensor model for such a sensor that includes at most one measurement per report. The reports are assumed to arrive irregularly, and they are processed sequentially. The presented sensor model is a bit different than the general sensor model suggested by Mahler.[4] Mahler suggests that the whole sensor suite is reconceptualised as a single "meta-sensor". In the model suggested in this paper, the sensors are asynchronous, and assumed to operate independent of each other. This is the situation, for example, if the sensors are connected to the fusion centre via an unknown delay and unreliable network connection.

2. THE RANDOM SET TRACKING MODEL

The multiple target tracking problem can be modelled in a very flexible and natural fashion with random sets. The unknown estimated multitarget state $\boldsymbol{X}_k$ is a random set* consisting of unknown and varying number of targets. Similarly, the measurements can be modelled as unknown-cardinality random sets $\boldsymbol{Y}_k$. Quite often, as well as in this paper, the state evolution $\boldsymbol{X}_{0:\infty}$ is modelled as a Markov chain. That is, the model is completely described by the Markov transition kernel, i.e. the conditional probability $P(\boldsymbol{X}_k \mid \boldsymbol{X}_{k-1})$. In this paper, the measurements $\boldsymbol{Y}_{1:\infty}$ are allowed to depend on the multitarget state sequence at one instantaneous time, and are assumed to be conditionally independent. In this case, the conditional probability $P(\boldsymbol{Y}_k \mid \boldsymbol{X}_k)$ describes

Further author information: E-mail: matti.vihola@datactica.fi

*In this paper, random elements are denoted with boldface letters. Uppercase letters denote sets. Random sets are denoted with bold uppercase letters. Indexed sequences are denoted so that, e.g., $X_{a:b} \triangleq \{X_j : a \leq j \leq b\}$.

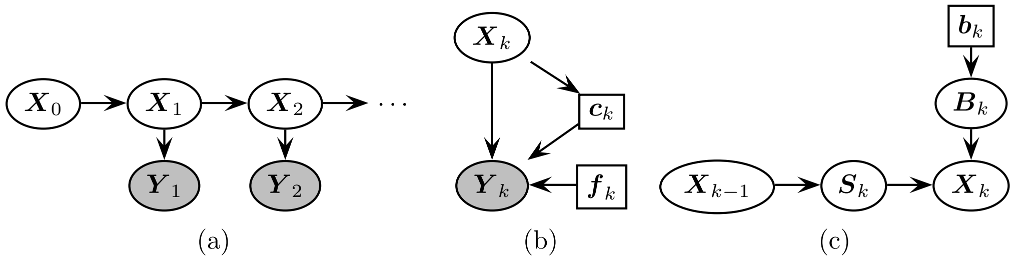


Figure 1. (a) Figure of a Bayesian network representing the random set tracking model. (b) Details of the measurement model $P(\boldsymbol{Y}_k \mid \boldsymbol{X}_k)$. (c) Details of the dynamic model $P(\boldsymbol{X}_k \mid \boldsymbol{X}_{k-1})$. The nodes representing the observed variables are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

the measurement model completely. This model is depicted as a Bayesian Network† (BN) in Fig. 1 (a). The construction of the two conditional distributions, $P(\boldsymbol{X}_k \mid \boldsymbol{X}_{k-1})$ and $P(\boldsymbol{Y}_k \mid \boldsymbol{X}_k)$, is described in Sects. 2.1 and 2.2.

Since the straightforward construction of the conditional probabilities for random sets is complicated, or even infeasible, some tools have been developed for the purpose. In the case of random vectors, the probabilistic models are usually constructed using either cumulative distribution functions (CDFs), or probability density functions (PDFs) of some well-known distributions. In the case of random sets, the belief measure $\beta_{\boldsymbol{X}} \triangleq P(\boldsymbol{X} \subset C)$ is a similar concept to the CDF of random vectors. The belief measure determines uniquely a random set measure. From the belief measure, one can constructively obtain the random set PDF $f_{\boldsymbol{X}}$. The belief measure and the density are related so that

$$\beta_{\boldsymbol{X}}(C) = \int_C f_{\boldsymbol{X}}(Z)\delta Z \tag{1}$$

for any (measurable) set $C \subset \mathbb{S}$. The integral in is the Eq. (1) is the *set integral.* The density function $f_{\boldsymbol{X}}(Z) = \delta\beta_{\boldsymbol{X}}/\delta Z$ is the *set derivative* of the belief density. The set integral, the set derivative as well as other concepts of the *finite set statistics* are described thoroughly in the book by Goodman et. al.[1] For such readers, who fancy general probability theory, the following references will provide a bit different view of the random set concepts.[6–8]

Since the random set approach is a Bayesian formulation, the core of the random set estimation is the Bayes recursion, which can be written for random set densities as follows,

$$p_{k|k-1}(X_k) = \int f_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}(X_k \mid X_{k-1}) p_{k-1|k-1}(X_{k-1})\delta X_{k-1} \tag{2}$$

$$p_{k|k}(X_k) = \frac{f_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(Y_k \mid X_k) p_{k|k-1}(X_k)}{\int f_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(Y_k \mid X'_k) p_{k|k-1}(X'_k)\delta X'_k} \tag{3}$$

where the notation $p_{a|b}(X) \triangleq f_{\boldsymbol{X}_a|\boldsymbol{Y}_{1:b}}(X \mid Y_{1:b})$ is used. The general Bayes recursion can be computed in closed form in very rare cases. The multitarget Bayes recursion is inherently intractable. This means, that one needs to develop a strategy how to approximate Eqs. (2) and (3). In this paper, the selected approach is the particle filter, i.e. a sequential Monte Carlo approximation of the posterior distribution.[9] The particle filter algorithm is described in Sect. 3.

2.1. The Dynamic Model

The multitarget random set $\boldsymbol{X}_k$ is assumed to be a finite subset of the state space $\mathbb{S}$‡. The target motions are assumed independent of each other. In addition, consider that the new target appearance and the old target

†The Bayesian networks are used only in order to illustrate the (in)dependencies in a probabilistic model. For further information on the Bayesian networks, see, e.g. the PhD thesis of K. Murphy.[5]

‡In the experiments of this paper, $\mathbb{S} = \mathbb{R}^4$, but the space can be virtually anything. The only restriction is that the space must be locally compact, Hausdorff, and separable.[1]

disappearance are independent both with respect to each other and with respect to the other targets' movements. Then, a multitarget dynamic model shown in Fig. 1 (c) can be constructed in parts,[10]

$$\boldsymbol{X}_k = \boldsymbol{S}_k \cup \boldsymbol{B}_k \tag{4}$$

where $\boldsymbol{B}_k$ is the set of the appeared new targets, and $\boldsymbol{S}_k$ is the set of the targets survived from the previous round $k-1$.

The dynamic model $P(\boldsymbol{S}_k \mid \boldsymbol{X}_{k-1})$ of the survived is defined by a belief measure $\beta_{\boldsymbol{S}_k}(C \mid X_{k-1}) = P(\boldsymbol{S}_k \subset C \mid \boldsymbol{X}_{k-1} = X_{k-1})$. Suppose that there are n targets on the set $X_{k-1} = \{x_{k-1}^{(1)}, \ldots, x_{k-1}^{(n)}\}$, then the set of the survived targets can be represented as[10]

$$\boldsymbol{S}_k = \bigcup_{i=1}^{n} \boldsymbol{X}_k^{(i)} \qquad \text{where} \qquad \boldsymbol{X}_k^{(i)} = \begin{cases} \{\boldsymbol{x}_k^{(i)}\} & \text{with probability } p_s(x_{k-1}^{(i)}) \\ \emptyset, & \text{with probability } 1 - p_s(x_{k-1}^{(i)}) \end{cases} \tag{5}$$

where $p_s : \mathbb{S} \to [0,1]$ characterises the probability of target survival in a given position of the state space.

Since $\boldsymbol{D}_k^{(i)}$ and $\{\boldsymbol{x}_k^{(i)}\}$ were assumed independent, the conditional belief measure of $\boldsymbol{X}_k^{(i)}$ can be given in the following form

$$\begin{aligned} \beta_{\boldsymbol{X}_k^{(i)}}(C \mid X_{k-1}) &= P(\boldsymbol{X}_k^{(i)} \subset C \mid \boldsymbol{x}_{k-1}^{(i)} = x_{k-1}) \\ &= P(\boldsymbol{X}_k^{(i)} = \emptyset \mid \boldsymbol{x}_{k-1}^{(i)} = x_{k-1}) + P(\{\boldsymbol{x}_k^{(i)}\} \subset C \mid \boldsymbol{x}_{k-1}^{(i)} = x_{k-1}) \\ &= [1 - p_s(x_{k-1})] + p_s(x_{k-1}) P_{\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}}(C \mid x_{k-1}) \end{aligned}$$

where $1 - p_s(x_{k-1})$ is the probability that a target in position x_{k-1} will disappear, and $P_{\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}}(C \mid x_{k-1}) = P(\boldsymbol{x}_k^{(i)} \in C \mid \boldsymbol{x}_{k-1}^{(i)} = x_{k-1})$ is the single-target dynamic model.

Next, one can ask how to obtain the belief measure corresponding $\beta_{\boldsymbol{S}_k}$. Since $\boldsymbol{X}_k^{(i)}$ were assumed independent, one obtains

$$\begin{aligned} \beta_{\boldsymbol{S}_k}(C \mid X) &= P\left(\bigcup_{i=1}^{n} \boldsymbol{X}_k^{(i)} \subset C \;\middle|\; \boldsymbol{X}_{k-1} = X \right) = \prod_{i=1}^{n} P(\boldsymbol{X}_k^{(i)} \subset C \mid \boldsymbol{X}_{k-1} = X) \\ &\prod_{i=1}^{n} P(\boldsymbol{X}_k^{(i)} \subset C \mid \boldsymbol{x}_{k-1}^{(i)} = x^{(i)}) = \prod_{i=1}^{n} \beta_{\boldsymbol{X}_k^{(i)}}(C \mid x^{(i)}) \end{aligned}$$

The set of new targets that have appeared $\boldsymbol{B}_k$ can have, in general, any distribution. A rather uninformative model can be constructed so that one assumes that the targets appear independently in different regions of the state space. That is, given that k targets are born, the elements $\boldsymbol{b}_k^{(i)}$ that constitute the set $\boldsymbol{B}_k$ are independent. This kind of Poisson birth model can be constructed piecewise as follows

$$\beta_{\boldsymbol{B}_k}(C \mid X_{k-1}) = \beta_{\boldsymbol{B}_k}(C) = \sum_{j=0}^{\infty} b_j P_b(C)^j, \quad \text{where} \quad b_j = \frac{(\eta\tau_k)^j}{j!} e^{-\eta\tau_k}, \quad j \in \mathbb{N} \tag{6}$$

where $P_b(C) = P(\boldsymbol{b}_k^{(i)} \in C)$ is a probability measure for the single target appearance and the terms $b_j = P(|\boldsymbol{B}_k| = j)$ denote the probability that exactly j targets are born. This target appearance model is a Poisson process. The term $\tau_k = t_k - t_{k-1}$ is the time difference between $\boldsymbol{X}_k$ and $\boldsymbol{X}_{k-1}$, and η is the birth intensity parameter, that describes the expected number of born targets per time unit.

Due to the independence of the individual target motions, the disappearance, and the appearance of the new targets, the conditional belief measure for tthe dynamic model in Fig. 1 (c) can be written as

$$\begin{aligned} \beta_{\boldsymbol{X}_k \mid \boldsymbol{X}_{k-1}}(C \mid X_{k-1}) &\triangleq P(\boldsymbol{X}_k \subset C \mid \boldsymbol{X}_{k-1} = X_{k-1}) = P(\boldsymbol{S}_k \cup \boldsymbol{B}_k \subset C \mid \boldsymbol{X}_{k-1} = X_{k-1}) \\ &= P(\boldsymbol{B}_k \subset C)\, P(\boldsymbol{S}_k \subset C \mid \boldsymbol{X}_{k-1} = X_{k-1}) = \beta_{\boldsymbol{B}_k}(C) \beta_{\boldsymbol{S}_k}(C \mid X_{k-1}) \end{aligned} \tag{7}$$

2.2. The Measurement Model

Suppose that there is a sensor that includes either one measurement or no measurements in each report. This section introduces a model for this type of sensor. The sensor model includes independent false alarms, that occur with probability p_f. In addition, the targets are assumed to be measured equiprobably. However, when a target is attempted to be measured, it may occur that the sensor fails to detect the target, thus there is also a model for the probability of detection. This measurement model can be factored into a BN shown in Fig. 1 (b). The binary random variable $\boldsymbol{f}_k$ can take the values 1 (a false alarm) and 0 (not a false alarm). The distribution of $\boldsymbol{f}_k$ can be given as follows.

$$P(\boldsymbol{f}_k = f) = \begin{cases} 1 - p_f, & f = 0 \\ p_f, & f = 1 \end{cases} \tag{8}$$

The association variable $\boldsymbol{c}_k$ is integer-valued, and has a distribution that depends only on the current number of targets, i.e. the cardinality of $\boldsymbol{X}_k$.

$$P\left(\boldsymbol{c}_k = i \,\middle|\, |\boldsymbol{X}_k| = n\right) = \begin{cases} 1/n, & 1 \le i \le n \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where it is assumed that $n \ge 1$. As we shortly see, the distribution of $\boldsymbol{c}_k$ is unimportant if $n = 0$. The measurement $\boldsymbol{Y}_k$ is assumed to depend on $\boldsymbol{X}_k$, $\boldsymbol{c}_k$, and $\boldsymbol{f}_k$, so that

$$\begin{aligned} \beta_{\boldsymbol{Y}_k}(C \mid \boldsymbol{X}_k, \boldsymbol{c}_k = i, \boldsymbol{f}_k = 1) &= P_f(C) \\ \beta_{\boldsymbol{Y}_k}(C \mid \boldsymbol{X}_k = \emptyset, \boldsymbol{c}_k = i, \boldsymbol{f}_k = 0) &= 1 \\ \beta_{\boldsymbol{Y}_k}(C \mid \boldsymbol{X}_k = X, \boldsymbol{c}_k = i, \boldsymbol{f}_k = 0) &= 1 - p_d(x^{(i)}) + p_d(x^{(i)})P(\boldsymbol{y} \in C \mid x^{(i)}) \end{aligned} \tag{10}$$

Now, the measurement model $P(\boldsymbol{Y}_k \mid \boldsymbol{X}_k)$ can be derived, since one can write

$$\begin{aligned} \beta_{\boldsymbol{Y}_k}(C \mid X) &= \sum_{f=0}^{1} \sum_{i=1}^{\infty} P(\boldsymbol{Y}_k \subset C, \boldsymbol{c}_k = i, \boldsymbol{f}_k = f \mid \boldsymbol{X}_k = X) \\ &= \sum_{f=0}^{1} \sum_{i=1}^{\infty} \left[\beta_{\boldsymbol{Y}_k}(C \mid \boldsymbol{X}_k = X, \boldsymbol{c}_k = i, \boldsymbol{f}_k = f) P(\boldsymbol{c}_k = i, \boldsymbol{f}_k = f \mid \boldsymbol{X}_k = X) \right] \\ &= \sum_{f=0}^{1} \sum_{i=1}^{\infty} \left[\beta_{\boldsymbol{Y}_k}(C \mid \boldsymbol{X}_k = X, \boldsymbol{c}_k = i, \boldsymbol{f}_k = f) P(\boldsymbol{c}_k = i \mid \boldsymbol{X}_k) P(\boldsymbol{f}_k = f) \right] \end{aligned} \tag{11}$$

Substituting Eqs. (8)–(10) into Eq. (11), one obtains the following belief measure $\beta_{\boldsymbol{Y}_k}(C \mid X) = P(\boldsymbol{Y}_k \subset C \mid \boldsymbol{X}_k = X)$.

$$\begin{aligned} \beta_{\boldsymbol{Y}_k}(C \mid \emptyset) &= (1 - p_f) + p_f P_f(C) \\ \beta_{\boldsymbol{Y}_k}(C \mid X) &= \frac{1 - p_f}{|X|} \left[\sum_{x \in X} \Big(1 - p_d(x)\Big) + \sum_{x \in X} \Big(p_d(x) P(\boldsymbol{y}_k \in C \mid x)\Big) \right] + p_f P_f(C) \end{aligned}$$

where $X \ne \emptyset$. The density corresponding to the belief measure above can be given as follows[§]

$$f_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(Y \mid X) = \begin{cases} 1 - p_f, & X = \emptyset, Y = \emptyset \\ p_f f_f(y), & X = \emptyset, Y = \{y\} \\ |X|^{-1}(1 - p_f) \sum_{x \in X} \left[1 - p_d(x)\right], & X \ne \emptyset, Y = \emptyset \\ |X|^{-1}(1 - p_f) \sum_{x \in X} \left[p_d(x) f_{\boldsymbol{y}_k|\boldsymbol{x}_k}(y \mid x)\right] + p_f f_f(y), & X \ne \emptyset, Y = \{y\} \end{cases} \tag{12}$$

where X denotes a nonempty set.

§For the derivation rules, consult e.g., the monograph by Mahler or the book by Goodman et. al.[1,4]

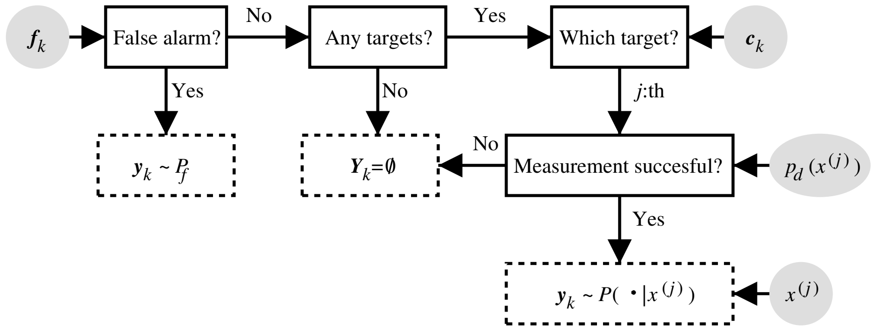


Figure 2. The data generation procedure for the model of zero or one measurements. The shaded ellipses denote the "inputs", while the dotted boxes contain the different possible outputs.

The derived measurement model can be considered intuitively through a procedure that generates a sensor report, outlined in Fig. 2. The generation of the report starts from the random variable $\boldsymbol{f}_k$ determining whether the current report contains a false alarm. If there is a false alarm, the measurement is distributed according to the false alarm distribution P_f. If the measurement is not a false alarm, i.e $\boldsymbol{f}_k = 0$, then the procedure continues with a query of any targets. If there are no targets, the measurement is empty. Otherwise, the i'th target is picked according to $\boldsymbol{c}_k = i$. The following query determines whether the sensor's attempt to obtain a measurement of the i'th target is a success. If the measurement attempt fails, the measurement is again empty. Otherwise, the measurement is generated according to the single-target measurement model $P(\boldsymbol{y}_k \in C \mid \boldsymbol{x}_k = x^{(i)})$.

3. THE PARTICLE FILTER IMPLEMENTATION

This section outlines a particle filter implementation of the random set tracking. The implementation covers the random set tracking model that was introduced in Sect. 2. The implementation is somewhat different to what Sidenbladh and Wirkander suggested.[3] Due to the different sensor model, the implementation admits a computational complexity that increases approximately linearly with respect to the expected number of targets. No prior limit for the number of targets is required, unlike in the algorithm proposed by Sidenbladh and Wirkander.[3] Vo et. al proposed this kind of an algorithm for general random set tracking in a conceptual level, but did not present any tests.[2]

The implementation that is described in this section is, in fact, exactly the general sampling importance resampling (SIR) filter[¶] with the adaptive resampling.[9, 11, 12] The proposed algorithm is outlined in Alg. 1. The symbols in the algorithm are in accordance with the symbols in Sects. 2.1 and 2.2. The effective sample size is computed so that[12] $n_{\text{eff}} = [\sum_{i=1}^{n} (\boldsymbol{w}_k^{(i)})^2]^{-1}$, where $\boldsymbol{w}_k^{(i)}$ are the particle weights. The resampling threshold n_{th} is an adjustable parameter. The symbol "$\sim$" can be read as "is a random sample distributed according to". Notice that the posterior expectation that is estimated has to be real or vector valued, so the function h maps finite sets to vectors.

The line of Alg. 1 (a) that states "$\boldsymbol{Z}_k^{(i)}$ is distributed according to $\beta_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}(\cdot \mid \boldsymbol{Z}_{k-1}^{(i)})$" is described in detail in Alg. 1 (b). The generation of the random samples $\boldsymbol{Z}_k^{(i)}$ follows directly the presentation in Sect. 2.1. At first, the number of born targets $\boldsymbol{m}$ is drawn from the discrete Poisson distribution. Then, the state of each of the born targets is drawn independently according to the birth distribution P_b. For each target in $\boldsymbol{Z}_{k-1}^{(i)}$, it is determined whether the target will survive. For the surviving targets, samples are drawn from the single-target motion model $P_{\boldsymbol{x}_k|\boldsymbol{x}_{k-1}}$.

4. OUTPUT AND VISUALISATION

Sect. 3 described the SMC random set tracking implementation. An important question is that how one can extract a reasonable estimate or estimates from the posterior distribution when it is approximated with particles.

[¶]Also known as the *bootstrap filter*.

$\boldsymbol{Z}_0^{(i)} \sim P_{\boldsymbol{X}_0}$
$\boldsymbol{w}_0^{(i)} \leftarrow 1/n$
for $k = 1, 2, \dots$ **do**
$\quad \boldsymbol{Z}_k^{(i)} \sim \beta_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}(\cdot \mid \boldsymbol{Z}_{k-1}^{(i)})$
$\quad \widehat{\boldsymbol{w}}_k^{(i)} \leftarrow f_{\boldsymbol{Y}_k|\boldsymbol{X}_k}(Y_k \mid \boldsymbol{Z}_k^{(i)})$
$\quad \boldsymbol{w}_k^{(i)} \leftarrow \dfrac{\widehat{\boldsymbol{w}}_k^{(i)}}{\sum_{i=1}^n \widehat{\boldsymbol{w}}_k^{(i)}}$
$\quad \mathsf{E}\left[h(\boldsymbol{X}_{0:k}) \mid \boldsymbol{Y}_{1:k} = Y_{1:k}\right] \approx \sum_{i=1}^n \boldsymbol{w}_k^{(i)} h(\boldsymbol{Z}_{0:k}^{(i)})$
$\quad$ **if** $n_{\text{eff}} < n_{\text{th}}$ **then**
$\quad\quad (\boldsymbol{Z}_k^{(i)}, \boldsymbol{w}_k^{(i)})_{i=1}^n \leftarrow \text{resample}\left[(\boldsymbol{Z}_k^{(i)}, \boldsymbol{w}_k^{(i)})_{i=1}^n\right]$
$\quad$ **end if**
end for

(a)

$\boldsymbol{m} \sim \text{Poisson}(\eta\tau_k)$
$\boldsymbol{B}_k \leftarrow \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_{\boldsymbol{m}}\}$, where $\boldsymbol{b}_i \sim f_b(\cdot)$
$\boldsymbol{S}_k \leftarrow \emptyset$
for all $\boldsymbol{z} \in \boldsymbol{Z}_{k-1}$ **do**
$\quad \boldsymbol{u} \sim U(0,1)$
$\quad$ **if** $\boldsymbol{u} < p_s(\boldsymbol{z})$ **then**
$\quad\quad \boldsymbol{S}_k \leftarrow \boldsymbol{S}_k \cup \{\boldsymbol{z}'\}$, where $\boldsymbol{z}' \sim P_{\boldsymbol{x}_k|\boldsymbol{x}_{k-1}}(\cdot \mid \boldsymbol{z})$
$\quad$ **end if**
end for
$\boldsymbol{Z}_k \leftarrow \boldsymbol{S}_k \cup \boldsymbol{B}_k$

(b)

Algorithm 1. (a) SIR implementation of the random set estimation framework. (b) Algorithm that draws a sample $\boldsymbol{Z}_k$ from the predictive distribution $\beta_{\boldsymbol{X}_k|\boldsymbol{X}_{k-1}}(\cdot \mid \boldsymbol{Z}_{k-1})$ given in Eq. (7). The symbol $U(0,1)$ stands for uniform $(0,1)$ distribution.

This means, in practice, that for which functions h in Alg. 1 the posterior expectation should be computed? The JoME and MaME estimates proposed by Mahler cannot be extracted from this representation straightforwardly, since both JoME and MaME are "MAP-like" estimates, which require maximisation of the posterior density.[4]

A reasonable estimator for the whole multitarget state is difficult to construct from the particle representation. Simple estimators can, however, be obtained for the number of targets. First, consider the functions

$$h_j(X) \triangleq \begin{cases} 1, & \text{if } |X| = j \\ 0, & \text{otherwise} \end{cases}$$

The cardinality distribution can be estimated using these functions as follows

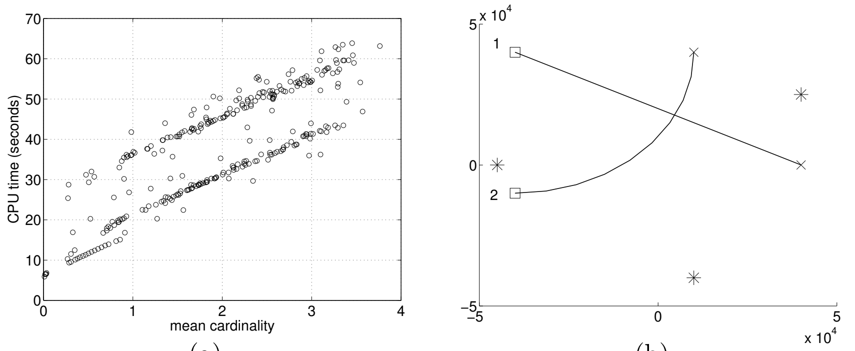
$$\widehat{p}(j) = \sum_{i=1}^n \boldsymbol{w}_k^{(i)} h_j(\boldsymbol{X}_k) = \sum_{i \in I(j)} \boldsymbol{w}_k^{(i)} \approx P(|\boldsymbol{X}_k| = j \mid \boldsymbol{Y}_{1:k} = Y_{1:k})$$

where $I(j) = \{i : |\boldsymbol{Z}_k^{(i)}| = j\}$. The expected *a posteriori* (EAP) and the maximum *a posteriori* (MAP) estimates for the number of targets can be extracted from this estimated cardinality distribution as follows

$$\widehat{n}_k^{\text{EAP}} = \sum_{j=0}^{\infty} \left[j \widehat{p}(j) \right], \qquad \widehat{n}_k^{\text{MAP}} = \arg\max_{j \in \mathbb{N}} \widehat{p}(j) \tag{13}$$

The infinite sum in Eq. (13) reduces naturally to a finite sum, since each particle contains a finite number of elements. Similarly, maximum has to be searched only from a finite set of possibilities. The EAP and MAP estimators for the number of targets differ also in the sense that $\widehat{n}_k^{\text{MAP}}$ is integer-valued, but $\widehat{n}_k^{\text{EAP}}$ is real-valued.

Since the position distribution of the targets is even more important than the cardinality distribution, it is clear that the position distribution has to be illustrated in some manner—preferably in a manner that is easy to interpret by a human. In the case of single-target tracking, a rather straightforward and intuitive visualisation of the single-target position distribution would be to consider a two-dimensional histogram corresponding the (x, y) coordinates. This type of histogram could be illustrated as an intensity image. In the multiple target case, similar information can be shown in the form of Probability Hypothesis Density (PHD). The PHD estimator that is considered in this paper is defined so that the surveillance region $\mathbb{S}$ is discretised into finite resolution



(a) (b)

Target ID	1	2
Start time	5	35
End time	220	270
Speed	416	891.1

Figure 3. (a) CPU time consumption for one processed measurement with respect to the mean cardinality of the particles. (b) The sensors (⊛) and the tracks (lines) in the scenarios. The tracks start from a square, and end into a cross. The table gives a summary of the temporal properties of the tracks.

cells. Then, for each finite resolution cell, the expected number of targets in the resolution cell is determined. This can be achieved as follows. Suppose $A \subset \mathbb{S}$ is a set corresponding to a resolution cell. Then, one determines the function $h_A(X) = |X \cap A|$ and obtains

$$\widehat{n}_k^{\mathrm{EAP}}(A) \triangleq \sum_{i=1}^{N} \boldsymbol{w}_k^{(i)} |\boldsymbol{Z}_k^{(i)} \cap A| \approx P(|\boldsymbol{X}_k \cap A| \mid \boldsymbol{Y}_{1:k} = Y_{1:k}) \tag{14}$$

The interpretation of the obtained PHD approximation is quite straightforward: the value $\widehat{n}_{k|k}^{\mathrm{EAP}}(A)$ of each resolution cell A corresponds to the (approximate) expectation of the number of targets within that cell.

5. EXPERIMENTS

The test scenarios consist of a varying number of targets in a surveillance region. There are sensors that measure the angle from the sensor towards the target based on some signal emitted by the target. This *bearings-only tracking* is a rather standard tracking scenario, that has been used in testing of single-target and multitarget tracking algorithms.[11, 13]

The experiments in this paper are based on artificial tracks. The measurement reports are generated according to the ideal measurement model, that is used also by the tracker. Since there are bearings-only measurements, which are corrupted by false alarms and missed measurements, the scenario can be considered difficult. The performance of the random set tracking algorithm was not compared with an existing tracking algorithm. This was due to the fact that no such alternative approaches were found in the literature that could be used in this kind of scenario.

Sect. 5.1 describes the details of the models that were used in the tests. In addition, the common parameters of the tracking scenarios are outlined. The specifics of the parameters of each test setup are given in Sect. 5.2, where the results are also analysed. More tests on this algorithm can be found in.[7]

5.1. Data and System Parameters

The algorithm was written in Matlab version 6.5. The tests were run on a PC having 2 GHz Intel Pentium 4 CPU and 512 MB of main memory. The number of Monte Carlo (MC) samples (particles) was fixed throughout the tests to 20000. The resampling threshold n_{th} was set to one fourth of the total number of particles, 5000. Fig. 3 (a) shows the computational load of the algorithm with respect to the mean cardinality of the particles. The measure of computational load is the elapsed CPU time (Matlab command `cputime`). The processing time is not a direct function of the mean cardinality, but the tendency can be seen: the processing load increases approximately linearly with respect to the mean cardinality.

The state space $\mathbb{S}$ in the test scenarios was four-dimensional consisting of the x and y-coordinates, and the x and y velocities, denoted as x' and y', respectively. The state space was bounded, so that for each $\underline{v} = [x, y, x', y']^T \in \mathbb{S}$, the following conditions were satisfied.[‖]

$$\begin{array}{rcccl} -50\text{km} & \leq & x, y & \leq & 50\text{km} \\ -1000\text{m/s} & \leq & x', y' & \leq & 1000\text{m/s} \end{array}$$

Fig. 3 (b) shows the positions of the three angular sensors, and the four constant speed tracks involved in the scenarios. All of the three sensors were used in all the tests, but the number of targets that were selected varied. The single-target scenario included only the target 1, and the two-target scenario included the targets 1 and 2.

If not otherwise mentioned, the measurement data was generated according to the measurement model involved in the corresponding test. All the scenarios had a time span from 1 to 300 seconds. The measurements arrived sequentially at each second. The measurement generation process was such that at each second, the sensor that produced the measurement was picked randomly. Then, a measurement was generated according to the sensor model, as illustrated in Fig. 2. Sect. 5.1.2 describes the common parameters of the measurement model, while the test-specific parameters are given in Sect. 5.2. The parameters of the dynamic model were fixed throughout the tests to the values described in Sect. 5.1.1.

5.1.1. The dynamic model

The single-target dynamic model was the constant velocity (or the white noice acceleration) model.[14] The process noise standard deviation ρ was fixed to the value 35 $\text{m/s}^{3/2}$. The value was set empirically. The initial distribution was constrained to be "no targets", i.e. $P(\boldsymbol{X}_0 = \emptyset) = 1$. The track birth rate η, i.e. the expected number of born targets per second, was set to value 0.1. The distribution of the born targets P_b was uniform over the whole surveillance region. For practical reasons, the Poisson distribution was truncated so that the maximum number of born targets between two processed sensor reports was one.

The probability of track survival was 0.95 at maximum. To prevent the targets getting out of the surveillance region, soft boundaries were created so that the probability of survival decreased linearly, if any of x, y is nearer than 10 km to the surveillance region border, or if any of x', y' is nearer than 50 m/s to the allowed velocity limits. When representing the target state as $\underline{v} = [x, y, x', y']^T$, the probability of survival can be given as follows

$$p_s(\underline{v}) = \begin{cases} \min\limits_i (0.95, \dfrac{\underline{v}(i) + \underline{M}(i)}{\underline{b}(i)}, \dfrac{\underline{M}(i) - \underline{v}(i)}{\underline{b}(i)}), & \underline{v} \in \mathbb{S} \\ 0, & \underline{v} \notin \mathbb{S} \end{cases}$$

$\underline{M} = 10^3[50, 50, 1, 1]^T$ and $\underline{b} = [10^4, 10^4, 50, 50]^T$. The case $\underline{v} \notin \mathbb{S}$, i.e. the vector is out of the surveillance region, is included due to the fact that the constant velocity single-target dynamic model allows the target to drift outside the surveillance region[**]. The probability of survival is exemplified in Fig. 4 (a) as a function of the (x, y)-position, with zero velocities.

5.1.2. The measurement model

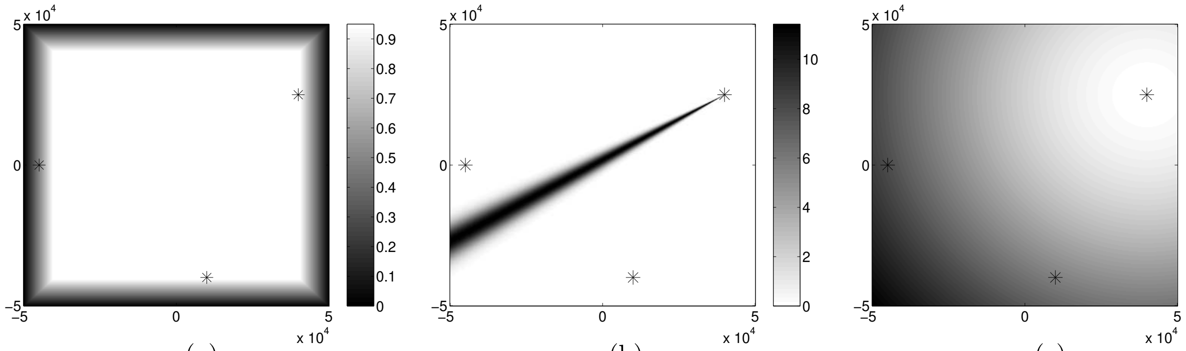
The sensors produced bearings-only measurements. This means, that the sensors returned only the direction of arrival (DOA) of the signal they measured. In the scenarios, the angular measurement model assumed additive Gaussian noise. This noisy DOA measurement model can be given as follows.

$$\boldsymbol{y}_k = \text{atan2}(\widehat{y}, \widehat{x}) + \boldsymbol{v}_k \mod 2\pi \tag{15}$$

where $\boldsymbol{v}_k$ is a Gaussian distributed random variable having standard deviation σ_a, $\widehat{x} = x - s_x$ and $\widehat{y} = y - s_y$ 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

[‖]Since it is convenient, for the sake of intuition, to introduce some units of measurement, we use the standard physical units of position and time. One should keep in mind, however, that the tests are purely synthetic.

[**]Strictly speaking, the state space should be considered as $\mathbb{S} = \mathbb{R}^4$. It is more convenient, however, to speak of the effective restriction of $\mathbb{R}^4$ as the state space.



(a) (b) (c)

Figure 4. (a) The probability of survival of a track (b) The single-target measurement likelihood function in the case of the measurement $y_k = 7\pi/6$ (c) The probability of detection for the same sensor. All the images illustrate the values of the functions with respect to the (x, y) position.

the four quadrant inverse tangent. Fig. 4 (b) illustrates the measurement model, showing an intensity image of the values of the measurement density function $f_{\boldsymbol{y}_k|\boldsymbol{x}_k}(a \mid [x, y, x', y']^T)$ with respect to x and y, when $a = 7\pi/6$.

The measurement model allowed also detections to be missed. For that purpose, the function $p_d(\underline{v})$ was defined to determine the probability of detection of a target position $\underline{v}$. The model that was used in the experiments was the following

$$p_d([x, y, x', y']^T) = \exp\left(-\frac{1}{2\sigma_r^2}\|[\widehat{x}, \widehat{y}]^T\|^2\right)$$

The sensor range parameter σ_r that was used in the experiments was 80 km. Fig. 4 (c) illustrates the values p_d gets in the different (x, y) positions of the state space. Finally, the sensor was allowed to produce false alarms. The test cover different false alarm rates (FARs), and the specific values are given below. The false alarm measurements were assumed to be distributed uniformly in $[0, 2\pi)$, i.e. $f_f(a) = (2\pi)^{-1}$ for all $0 \leq a < 2\pi$.

5.2. Tests

The outputs of the tracking algorithm that were included in the analysis consisted of the two estimates for the number of targets given in Eq. (13). The PHD visualisation was carried out according to the estimator given in Eq. (14). The surveillance region was divided into 20 equiwidth intervals with respect to x and y dimensions, determining 400 equivolume (x, y) resolution cells. The velocity resolution was one, i.e. the velocity components were not distinguished in any manner.

The actual tests consisted of three setups. The first setup analysed in Sect. 5.2.1 outlines the effect of increased FARs, while Sect. 5.2.2 contains a test which determines how the algorithm can handle with a mismatch between the true value of false alarm parameter and the parameter value given to the algorithm.

5.2.1. False alarm rate

The first test setup consisted of test runs with the single-target and the two-target scenario, with three different increased FARs. The results for the single-target scenario with FARs 30%, 70%, and 80% are shown in Fig. 5. The top row of Fig. 5 shows the development of the estimators for the number of targets. The uppermost graph shows the development of the EAP estimator (the dotted line), and the true number of targets (the solid line). The two graphs below show the development of errors of the EAP and MAP estimators, i.e. the differences between the values of the estimators and the true target count. The bottom row of Fig. 5 shows a snapshot of the PHD approximation at one time instant, after processing the 111th measurement. The star (☆) symbol in the PHD image corresponds to the true position of a target. Notice the logarithmic scale of the intensities in the PHD image. The absolute value of a particular bin of PHD is generally of no great importance. The general shape of the PHD image and the current estimated number of targets are more important. The numerical value of the EAP estimator is shown in the title of the PHD image in addition to the current time instant in seconds.

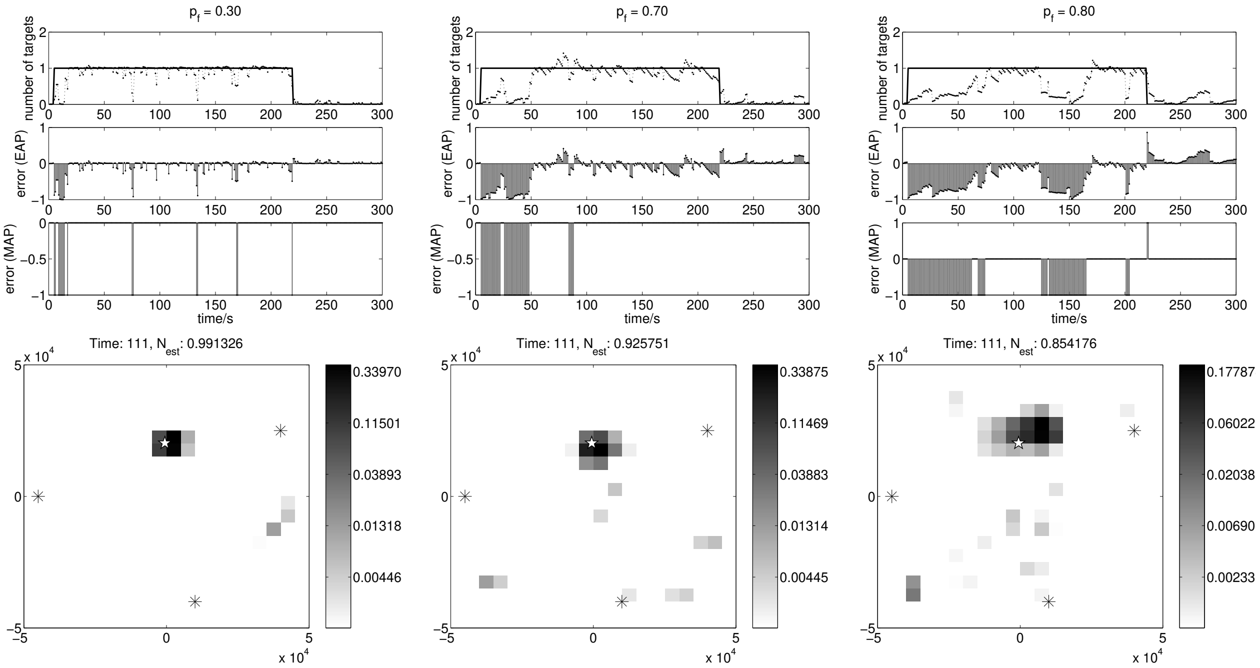


Figure 5. Top: Development of the estimates of the number of targets in the single-target scenario, with different probability of false alarm levels. Bottom: PHD snapshots after processing of the 111th measurement.

The PHD snapshots in Fig. 5 show how the algorithm reacts to the false alarms. There are "ghost" tracks in addition to the true one. Due to the increased FAR, there are less true target measurement, which delays the initiation of the track. This can be seen in Fig. 5, where the EAP and MAP estimates rise into unity approximately after the 15th measurement in the case of 30% FAR, but only after the 50th measurement in the case of 70% FAR. According to this setup, the algorithm can handle false alarms rather well, providing quite reliable tracking results with as high FAR as 70% for single target. As FAR is increased to 80%, the algorithm seems to lose the track temporarily.

The algorithm was tested also in the two-target scenario, with three increased FARs: 10%, 30%, and 50%. The development of the estimates of the number of targets are shown in Fig. 6. As expected, the performance of the algorithm decreases more rapidly in the two-target scenario, when the FAR is increased. It must be mentioned that in this test setup, as FAR is increased the number of the true measurements from each target decreases.

5.2.2. Parameter mismatch

In real world, the tracking model rarely matches the characteristics of the true tracking environment. Most importantly, the parameters of the model that are adjusted manually can be expected to be misleading in some cases. So, a tracking framework should be robust against incorrect parameter values, at least in some extent[††]. Therefore, a test of robustness against a parameter mismatch was included. The target parameter was the false alarm rate.

The test was carried out using the single-target scenario. The only parameter value that was altered was the probability of false alarm, p_f. The FARs that were used in the experiment were 5%, 30% and 70%. The input data, i.e. the measurements, were generated separately according to each FAR. After that, for each input data, the algorithm was run with each of the three values of the probability of false alarm parameter p_f.

[††]Of course, if an algorithm is made extremely robust against incorrect parameter values, then the parameters do not affect in any way.

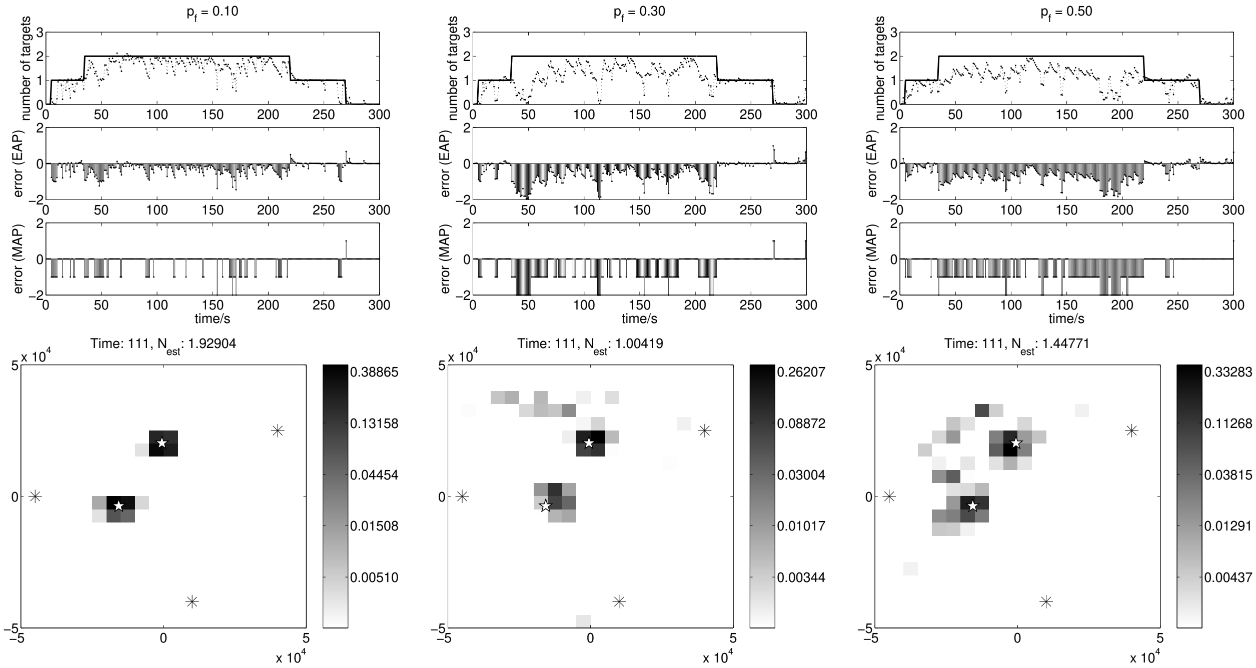


Figure 6. Top: the development of the estimates of target count in the two-target scenario with different FARs. Bottom: the corresponding PHD snapshots after processing of the 111th measurement.

Table 1. Mean absolute error of the estimates of the number of targets, when the parameter value was altered from the true ones. The errors of the EAP estimate are shown in (a), and the errors of the MAP estimate in (b). The smallest error value in each row is shown in boldface.

(a)

True value	Parameter value 5%	30%	70%
5%	0.077	**0.075**	0.106
30%	0.104	**0.086**	0.107
70%	0.581	0.244	**0.214**

(b)

True value	Parameter value 5%	30%	70%
5%	**0.047**	0.050	0.087
30%	**0.040**	0.053	0.073
70%	0.480	0.180	**0.153**

The mean absolute errors of the EAP and MAP estimates in the case of all the combinations of the parameter values and the true (simulated) values are shown in Table 1. The trend seems to be that the true parameter values produced the best results. However, for 5% FAR input data, the mean absolute error of the EAP for algorithm running on 30% setting was the smallest. Similarly, for 30% FAR input data, the algorithm running on p_f setting 5% had a smaller mean MAP error than the true setting. These "anomalies" may be explained by the Monte Carlo variation in the output of the algorithm, and the particular input data. Most importantly, though, the performance of the algorithm decreased gradually with respect to the mismatch in the parameter value. The worst result was obtained with input data having 70% false alarms, and the parameter value set to 5%.

6. CONCLUSIONS AND FUTURE WORK

This paper described a SMC random set tracking implementation in the case of independently operating sensors producing at most one target measurement in each report. The SMC implementation was tested in simple but challenging bearings-only tracking scenarios. The experimental results show that the random set framework has potential in such a challenging tracking task. The algorithm showed robustness against relatively high false alarm rates (FARs). The degradation of the performance of the algorithm was gradual with respect to the increasing FAR. This property is promising in view of low-fidelity sensors that are prone to produce many false alarms. The algorithm was tested also with incorrectly set FAR parameter values. The performance of the algorithm degraded gradually as the mismatch increased.

It seems that, without substantial reduction of the computational complexity, the SMC implementation of the random set Bayes recursion cannot be applied to tracking of a large number (tens or more) of targets. The computational complexity of the algorithm increases approximately linearly with respect to the number of targets, when the number of Monte Carlo (MC) samples is kept fixed. In practice, however, the number of MC samples needs to be increased as the number of targets increases in order to achieve a given accuracy. Hence, the computational complexity of the algorithm increases more rapidly in practice. For such purposes, when there are a lot of targets, the PHD approximation of the random set tracking framework may be considered a better alternative.

In future, the algorithm could be made more effective by selecting a different importance distribution. The predictive importance distribution used by the described SIR filter is quite ineffective. Especially such a choice of an importance distribution that takes the measurements into account in drawing the states of the born targets could be considered. The Rao-Blackwellised SMC has been proposed to tracking multiple targets with bearings-only measurements.[13] It is possible to derive a similar approach in the random set framework. The PHD approximation[10] could be researched also in the case of the proposed measurement model.

REFERENCES

1. I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, vol. 39 of *Series B: Mathematical and Statistical Methods*, Kluwer Academic Publishers, AA Dordrecht, The Netherlands, 1997.
2. B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proceedings of the Sixth International Conference on Information Fusion*, **2**, pp. 792–799, (Cairns, Australia), July 2003.
3. 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.
4. R. Mahler, *An Introduction to Multisource-Multitarget Statistics and its Applications*, Lockheed Martin, Mar. 2000.
5. K. P. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, University of California, Berkeley, 2002.
6. B.-N. Vo and S. Singh, "On the Bayes filtering equations of finite set statistics," in *The Proceedings of the 5th Asian Control Conference*, pp. 1273–1278, (Melbourne, Australia), 2004.
7. M. Vihola, *Random Sets for Multitarget Tracking and Data Fusion.* Licentiate thesis, Tampere University of Technology, Tampere, 2004. Available at `http://iki.fi/mvihola/publications.html`.
8. B.-N. Vo, S. S. Singh, and A. Doucet, "Sequential Monte Carlo methods for Bayesian multi-target filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems* , 2004. (accepted).
9. A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.
10. R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems* **39**, pp. 1152–1178, Oct. 2003.
11. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F* **140**(2), pp. 107–113, 1993.
12. A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR.310, Signal Processing Group, Department of Engineering, University of Cambridge, Cambridge, UK, 1998.
13. S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-Blackwellized Monte Carlo data association for multiple target tracking," in *Proceedings of the Seventh International Conference on Information Fusion*, pp. 583–590, (Stockholm, Sweden), June 2004.
14. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, San Diego, 1988.