Sensor Control for Multi-Object State-Space Estimation Using Random Finite Sets ⋆

Branko Ristic [a], Ba-Ngu Vo [b]

[a] *ISR division, DSTO, 506 Lorimer Street, Melbourne VIC 3207, Australia*

[b] *Department Elect. & Electronic Eng., The University of Melbourne, VIC 3010, Australia*

Abstract

The problem addressed in this paper is information theoretic sensor control for recursive Bayesian multi-object state-space estimation using random finite sets. The proposed algorithm is formulated in the framework of partially observed Markov decision processes where the reward function associated with different sensor actions is computed via the Rényi or alpha divergence between the multi-object prior and the multi-object posterior densities. The proposed algorithm in implemented via the sequential Monte Carlo method. The paper then presents a case study where the problem is to localise an unknown number of sources using a controllable moving sensor which provides range-only detections. Four sensor control reward functions are compared in the study and the proposed scheme is found to perform the best.

Key words: Sensor management, Bayesian estimation, random finite sets, particle filter, sequential Monte Carlo estimation, information measure

1 Introduction

The theory of Bayesian multi-object recursive state-space estimation (or filtering), was developed by Mahler [15] in the framework of finite set statistics (FISST). In order to deal with situations where both the number of objects and their positions in the state space are random and unknown, a multi-object state is modelled by a random finite set (RFS). Due to the imperfections of sensors (e.g. objects are detected only with certain probability $p_D \leq 1$; detections typically include false alarms), the acquired measurement sets are also modelled as random finite sets. Using the mathematical tools of the FISST, Mahler generalised the well-known Bayesian state-space estimation recursions for a single object [8, Ch.6], to their multi-object counterparts. While the generalisation is not trivial, it provides a direct analogy to single-object system filtering and control.

The problem addressed in the paper is sensor control (a.k.a. sensor management) for Bayesian multi-object filtering. Typically surveillance sensors are capable of a variety of actions, such as looking in certain directions, moving to other locations, using different modes of operation (waveform, beam-pattern), etc. The objective of sensor management is the on-line control of individual sensors so that the overall utility of the surveillance system is maximised. Sensor management thus represents sequential decision making, where each decision generates observations that provide additional information. Decisions are made in the presence of uncertainty both in the state space and in the measurement space. The assumption is that past decisions and past observations are available when making the next decision. This class of problems has been studied in the framework of partially observed Markov decision processes (POMDPs) [4]. The elements of a POMDP include the (uncertain) current information state, a set of admissible sensor actions and the reward function associated with each action. We adopt the information theoretic approach to sensor management, where the uncertain state is represented by a probability density function, while the reward function is a measure of the *information gain* associated with each action.

Sensor management in the context of Bayesian multi-object filtering has been considered earlier in a series of publications by Mahler, Zatezalo, El-Fallah and others. Mahler put forward the idea of using the Kullback-

⋆ This paper was not presented at any IFAC meeting. A short 4-page version appeared at COGIS 09 conference, Paris, France. Corresponding author B. Ristic, tel. (+61 3) 9626 8226, fax (+61 3) 9626 8341.

Email addresses: branko.ristic@dsto.defence.gov.au (Branko Ristic), bnvo@unimelb.edu.au (Ba-Ngu Vo).

Leibler discrimination as the reward function for multi-target sensor management in [13], but did not discuss how to implement it. Later he developed theoretical foundations of the multi-target sensor management reward function referred to as the posterior expected number of targets (PENT) [14,16]. Applications of PENT to the surveillance of the outer space are presented in [24,6]. Witkoskie *et al.* [23] applied the Bayesian multi-object filter implemented via a Gaussian mixture approximation, in conjunction with the "restless bandit" approach to sensor management, to track moving vehicles in a road network.

In this paper we propose and analyse an algorithm for sensor management in association with the Bayesian multi-object filter, formulated in the framework of partially observed Markov decision processes. The reward function associated with different sensor actions is computed via the Rényi or alpha divergence between the multi-object prior and the multi-object posterior FISST probability densities. The computation of the reward function is carried out by the sequential Monte Carlo method. Although the proposed approach has some similarity with sensor management using joint multi-target particle filtering [11], it is more general than [11] being formulated in the RFS framework which can deal with both false alarms and missed detections. The paper demonstrates the benefits of the proposed sensor management scheme via a case study, where the goal is to localise an unknown number of sources using range-only measurements collected by a controllable moving sensor.

The paper is organised as follows. Section 2 reviews the multi-object Bayes filter. Sensor management is described in Section 3, while the sequential Monte Carlo implementations of both the filter and the reward function are presented in Section 4. Section 5 describes the case study and its numerical results. Finally the conclusions are drawn in Section 6.

2 Multi-object Bayes filter

Suppose that at time k there are n_k object states $\mathbf{x}_{k,1},\ldots,\mathbf{x}_{k,n_k}$, each taking values in a state space $\mathcal{X}\subseteq\mathbb{R}^{n_x}$, and m_k measurements (detections) $\mathbf{z}_{k,1},\ldots,\mathbf{z}_{k,m_k}$, each taking values in the observation space $\mathcal{Z}\subseteq\mathbb{R}^{n_z}$. A multi-object state and a multi-object observation are then represented by the finite sets:

$$\mathbf{X}_k=\{\mathbf{x}_{k,1},\ldots,\mathbf{x}_{k,n_k}\}\in\mathcal{F}(\mathcal{X}), \quad (1)$$
$$\mathbf{Z}_k=\{\mathbf{z}_{k,1},\ldots,\mathbf{z}_{k,m_k}\}\in\mathcal{F}(\mathcal{Z}), \quad (2)$$

respectively. Here $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$ are the finite subsets of $\mathcal{X}$ and $\mathcal{Z}$, respectively. Due to imperfections of detectors, it is possible that at time k some of the targets in $\mathbf{X}_k$ are not detected. Moreover, the observation set $\mathbf{Z}_k$ typically includes false detections (or clutter) in addition to target originated detections. Random finite sets are used to model uncertainty in the multi-object state and multi-object measurement.

The objective of the recursive Bayesian multi-object state-space estimator [15] is to determine at each time step k the FISST posterior probability density [1] of multi-object state $f_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k})$, where $\mathbf{Z}_{1:k}=(\mathbf{Z}_1,\ldots,\mathbf{Z}_k)$ denotes the accumulated observation set sequence up to time k. The multi-object posterior can be computed sequentially via the prediction and the update steps. Suppose that $f_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k})$ is known and that a new set of measurements $\mathbf{Z}_{k+1}$ corresponding to time $k+1$ has been received. Then the predicted and updated multi-object posterior densities are calculated as follows [15]:

$$f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) = \int \pi_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k)\, f_k(\mathbf{X}_k|\mathbf{Z}_{1:k})\delta\mathbf{X}_k \quad (3)$$

$$f_{k+1}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k+1}) = \frac{g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1})f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k})}{\int g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X})f_{k+1|k}(\mathbf{X}|\mathbf{Z}_{1:k})\delta\mathbf{X}}, \quad (4)$$

where $\pi_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k)$ is a multi-object transition density and $g_k(\mathbf{Z}_k|\mathbf{X}_k)$ is a multi-object likelihood. We should note that this recursion is a non-trivial generalization, since the transition density needs to consider the uncertainty in target number, which can change over time due to targets entering and leaving the state space, and the multi-object likelihood needs to consider detection uncertainty and false alarms. It is also clear that the integrals in the recursion (3)-(4) are non-standard and need the notion of a *set integral*. Let $f(\mathbf{Y})$ be a real-valued function of a RFS $\mathbf{Y}$. Then its set integral is a sum of conventional integrals over all different cardinalities of $\mathbf{Y}$ [15, p.361]:

$$\int f(\mathbf{Y})\delta\mathbf{Y}\triangleq\sum_{n=0}^{\infty}\frac{1}{n!}\int f(\{\mathbf{y}_1,\ldots,\mathbf{y}_n\})d\mathbf{y}_1,\cdots d\mathbf{y}_n \quad (5)$$

Similarly to the single-object Bayes filter, the recursions (3)-(4) in general have no analytic closed-form solution. The most popular implementation of the Bayes multi-object filter is then based on the sequential Monte Carlo (SMC) method [21], where a set of weighted particles is recursively propagated as an approximation of the multi-object posterior density.

[1] While the FISST densities are not probability densities, they have been shown to be equivalent to probability densities on $\mathcal{F}(X)$ relative to some reference measure [21]. Subsequently, we do not distinguish between FISST densities and probability densities in this paper.

3 Sensor control

Sensor control will be carried out in the framework of a partially observed Markov decision process (POMDP) [4]. The elements of a POMDP include:

- an information state represented by the multi-object posterior pdf;
- a control vector (or action) belonging to a set of admissible controls, and
- a reward function associated with different control vectors.

Let $\mathbf{u}_k(i_k) \in \mathbb{U}_k(i_k)$ denote the control vector applied at time t_k to observer $i_k \in \{1, \ldots, O\}$ where $\mathbb{U}_k(i)$ is the set of admissible control vectors at time t_k for observer i. The notation should be extended to include the applied control vectors so that $f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1})$ is the posterior pdf after applying the control vector sequence $\mathbf{u}_0(i_0), \ldots, \mathbf{u}_{k-1}(i_{k-1})$ and observing the measurement set sequence $\mathbf{Z}_{1:k}$. After processing the first k measurement sets, the control vector $\mathbf{u}_k(i_k)$ to be applied at time t_k to observer i_k is selected as

$$\mathbf{u}_k = \arg\max_{\mathbf{v}\in\mathbb{U}_k} \mathbb{E}[\mathcal{D}(\mathbf{v}, f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1}), \mathbf{Z}_{k+1}(\mathbf{v}))] \tag{6}$$

where $\mathcal{D}(\mathbf{v}, f, \mathbf{Z})$ is the real-valued reward function associated with the control $\mathbf{v}$, at the time when the information state is represented by multi-object pdf f and when the application of control $\mathbf{v}$ would result in the (future) measurement set $\mathbf{Z}$. The fact that the reward function $\mathcal{D}$ depends on the future measurement set is undesirable, since we want to decide on future actions without actually applying them before the decision is made. Hence in (6) the expectation operator $\mathbb{E}$ is taken with respect to the prior measurement set pdf:

$$p_{k+1|k}(\mathbf{Z}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{v}) = \int g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}, \mathbf{v}) f_{k+1|k}(\mathbf{X}|\mathbf{Z}_{1:k}) \delta\mathbf{X}. \tag{7}$$

where $\mathbf{v} \in \mathbb{U}_k$.

We point out that eq. (6) maximises the reward based on a single future step only (the "myopic" policy). Ideally, sensor management policy would have to look an infinite number of steps ahead. While conceptually it is straightforward to generalise (6) to the multiple-step ahead reward function, the computational cost would grow exponentially with the number of steps ahead. Hence all practical multi-step ahead solutions are based on approximations or heuristics [1].

Reward functions are typically based on a decrease of uncertainty or *information gain*, measured from the information state. The information gain can be formulated using various information measures, such as the Fisher information, entropy, Kullback-Leibler (KL) divergence, etc [7]. We will adopt the reward function based on the Rényi (or alpha) divergence, which measures the information gain between the current and the future information state. The Rényi divergence between any two densities, $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$, is defined as [7,10]:

$$I_\alpha(p_1, p_0) = \frac{1}{\alpha - 1} \log \int p_1^\alpha(\mathbf{x}) p_0^{1-\alpha}(\mathbf{x}) d\mathbf{x} \tag{8}$$

where $\alpha \geq 0$ is a parameter which determines how much we emphasize the tails of two distributions in the metric. In the special cases of $\alpha \to 1$ and $\alpha = 0.5$, the Rényi divergence becomes the Kullback-Leibler divergence and the Hellinger affinity [18], respectively [7].

The reward function $\mathcal{D}(\mathbf{v}, f, \mathbf{Z})$ in (6) is then the Rényi divergence between the predicted multi-object posterior density $f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1})$ specified by (3), and the future updated posterior multi-object density of (4), $f_{k+1}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k+1}, \mathbf{u}_{0:k})$, which is computed using the new measurement set $\mathbf{Z}_{k+1}$, obtained after sensor i_k has been controlled to take action $\mathbf{u}_k(i_k)$. According to (8) the reward function can be written as (in order to simplify notation we suppress the second and third argument of $\mathcal{D}$):

$$\mathcal{D}(\mathbf{u}_k) = \frac{1}{\alpha - 1} \log \int [f_{k+1}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1}, \mathbf{Z}_{k+1}, \mathbf{u}_k)]^\alpha [f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1})]^{1-\alpha} \delta\mathbf{X}_{k+1} \tag{9}$$

To further simplify notation, we drop $\mathbf{u}_{0:k-1}$. Next we substitute (4) into (9) to obtain:

$$\mathcal{D}(\mathbf{u}_k) = \frac{1}{\alpha - 1} \log \Big\{ \frac{1}{[p_{k+1|k}(\mathbf{Z}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_k)]^\alpha} \int [g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1}, \mathbf{u}_k)]^\alpha f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) \delta\mathbf{X}_{k+1} \Big\} \tag{10}$$

where $p_{k+1|k}(\mathbf{Z}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_k)$ was defined in (7). Recall from (6) that we require the expected value of the reward function in (10) to select the control vector.

Since the analytic solutions to both multi-object nonlinear filtering recursions (3)-(4) and the expected reward function $\mathbb{E}[\mathcal{D}(\mathbf{u}_k)]$ do not exist, we need to employ numerical approximations. We adopt the sequential Monte Carlo method [5,19].

4 SMC Implementation

The multi-object posterior $f_k(\mathbf{X}_k|\mathbf{Z}_{1:k})$ can be approximated by a set of random samples (or multi-object particles) $\{\mathbf{X}_k^i\}_{i=1}^N$, where each multi-object

particle $\mathbf{X}_k^i = \{\mathbf{x}_{k,1}^i, \ldots, \mathbf{x}_{k,n_k^i}^i\}$ postulates both the the number of objects in the surveillance volume $n_k^i = |\mathbf{X}_k^i| \in \{0, 1, 2, \ldots\}$, and their positions (support points) in the state space, $\mathbf{x}_{k,j}^i$, $j = 1, \ldots, n_k^i$. Let $\{w_k^i\}_{i=1}^N$ be the weights associated to the particles. Then we can write [15, Sec.15.3]:

$$f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}) \approx \sum_{i=1}^{N} w_k^i \, \delta_{\mathbf{X}_k^i}(\mathbf{X}_k). \tag{11}$$

where $\delta_{\mathbf{Y}}(\mathbf{X})$ is the multi-object Dirac delta function defined as follows [15, p.366]. Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be two sets, then $\delta_{\mathbf{Y}}(\mathbf{X}) = 0$, if $m \neq n$, $\delta_{\mathbf{Y}}(\mathbf{X}) = 1$ if $n = m = 0$ and

$$\delta_{\mathbf{Y}}(\mathbf{X}) = \sum_{\pi \in \Pi_n} \delta(\mathbf{x}_1 - \mathbf{y}_{\pi(1)}) \cdots \delta(\mathbf{x}_n - \mathbf{y}_{\pi(n)}) \tag{12}$$

if $n = m > 0$. The summation in (12) is over the set of all permutations Π_n of $\{1, \ldots, n\}$. The weights w_k^i in (11) are positive and sum to one. As $N \to \infty$, the approximation (11) approaches the true multi-object posterior.

The predicted particles at time $k+1$ can be formed as [21]:

$$\mathbf{X}_{k+1|k}^i \sim q_{k+1|k}(\mathbf{X}_{k+1|k}|\mathbf{X}_k^i, \mathbf{Z}_k) \tag{13}$$

where $q_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k^i, \mathbf{Z}_k)$ is the importance sampling density. An inefficient but convenient choice for it is the multi-object transitional pdf $\pi_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k^i)$. The predicted multi-object pdf is approximated as:

$$f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) \approx \sum_{i=1}^{N} w_k^i \, \delta_{\mathbf{X}_{k+1|k}^i}(\mathbf{X}_{k+1}) \tag{14}$$

The SMC implementation of the multi-object nonlinear filter is discussed in [21]. Here we focus on the reward function for sensor control. Upon substitution of (14) into (10), we obtain:

$$\mathcal{D}(\mathbf{u}_k) \approx \frac{1}{\alpha - 1} \log \frac{\sum_{i=1}^{N} w_k^i \, [g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1|k}^i, \mathbf{u}_k)]^\alpha}{\left[\sum_{i=1}^{N} w_k^i \, g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1|k}^i, \mathbf{u}_k)\right]^\alpha} \tag{15}$$

Next we compute the expectation of the reward (15) with respect to the multi-object prior measurement pdf $p_{k+1|k}(\mathbf{Z}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_k)$. Again we use Monte Carlo approximation to achieve this.

Let us introduce

$$\gamma_\alpha(\mathbf{Z}_{k+1}|\mathbf{u}_k) = \sum_{i=1}^{N} w_k^i \, [g_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1|k}^i, \mathbf{u}_k)]^\alpha. \tag{16}$$

Note that for $\alpha = 1$, $\gamma_1(\mathbf{Z}_{k+1}|\mathbf{u}_k)$ is the Monte Carlo approximation of $p_{k+1|k}(\mathbf{Z}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_k)$ of (7). Then we can approximate the expectation of the reward function as follows:

$$\mathbb{E}[\mathcal{D}(\mathbf{u}_k)] \approx \sum_{j=1}^{M} \gamma_1(\mathbf{Z}_{k+1}^j|\mathbf{u}_k)\mathcal{D}(\mathbf{u}_k) \tag{17}$$

$$= \frac{1}{\alpha - 1} \sum_{j=1}^{M} \gamma_1(\mathbf{Z}_{k+1}^j|\mathbf{u}_k) \log \frac{\gamma_\alpha(\mathbf{Z}_{k+1}^j|\mathbf{u}_k)}{\gamma_1(\mathbf{Z}_{k+1}^j|\mathbf{u}_k)^\alpha} \tag{18}$$

where $\mathbf{Z}_{k+1}^j$, $j = 1, \ldots, M$ are random set samples from $p_{k+1|k}(\mathbf{Z}_{k+1}|\mathbf{Z}_{1:k}, \mathbf{u}_k)$ approximated by $\gamma_1(\mathbf{Z}_{k+1}|\mathbf{u}_k)$. Expression (18) converges to the true expected value of the reward function as $M, N \to \infty$.

5 A case study

This section presents a numerical demonstration of the proposed approach. For simplicity and without any loss of generality we focus on the update step, because the sensor control is required only for the purpose of collecting the observation set used in the filter update. Thus we consider only a constant number of static objects, thus ignoring the prediction step. We point out, however, that the proposed approach is applicable to any situation described in Sec.2 (moving objects, appearing/disappearing objects), but a more complicated study would only obscure the results and conclusions of a comparative analysis.

5.1 Problem description

Suppose an unknown number of objects $n \geq 0$ are present in a specified surveillance area $\mathcal{A}$. The unknown state of each object $r = 1, \ldots, n$ is fully specified by its location in the Cartesian coordinates (x, y), i.e. the single-object state space $\mathcal{X} \equiv \mathcal{A} \subseteq \mathbb{R}^2$ is two-dimensional and the state vector of object r is $\mathbf{x}_r = [x_r \; y_r]^T$.

We have at our disposal controllable moving sensors (observers) which provide noisy range-only scalar measurements $z \in \mathbb{R}$ (e.g. a range radar) with the probability of detection $p_D(\mathbf{x})$ and with a certain false alarm rate. The false alarms (clutter) are modelled as a Poisson RFS: their cardinality distribution (a discrete probability distribution of the number of elements in the RFS) is Poisson with mean λ; for a given cardinality, the false alarms are each iid with the spatial pdf $c(z|\mathbf{x})$. If an object r ($r = 1, \ldots, n$) is detected at time k by sensor i_k located at $\mathbf{u}_k(i_k) = [\chi_k^{i_k} \; \varphi_k^{i_k}]^T$, the received measurement is modelled by:

$$z_k^{i_k} = \| \, \mathbf{x}_r - \mathbf{u}_k(i_k) \, \| + v_k^{i_k} \tag{19}$$

where $\| \mathbf{y} \|$ is the norm of vector $\mathbf{y}$, $v_k^{i_k}$ is zero-mean white Gaussian measurement noise, $v_k^i \sim \mathcal{N}(v; 0, \sigma_{k,i}^2)$.

In simulations the specified area $\mathcal{A}$ is a square $\mathcal{A} = [0, 1000] \times [0, 1000]\text{m}^2$. The probability of detection is homogeneous and constant across $\mathcal{A}$, $p_D = 0.95$, the mean number of false detections is $\lambda = 0.8$, and the spatial distribution of false detections is uniform $c(z|\mathbf{x}) = c(z) = \mathcal{U}[0, R_{\max}]$ with $R_{\max} = \sqrt{2} \cdot 1000\text{m}$. All these parameters are known to the estimation algorithm. Furthermore, we 'place' $n = 2$ objects in $\mathcal{A}$ at $\mathbf{x}_1 = (300, 600)\text{m}$ and $\mathbf{x}_2 = (750, 250)\text{m}$. For the duration of the simulation, the number of objects and their locations remain constant. The algorithm will provide sequentially the point estimates $\hat{\mathbf{X}}_k$ (which translate to $\hat{n}_k$, $\hat{\mathbf{x}}_{k,1}, \ldots, \hat{\mathbf{x}}_{k,\hat{n}_k}$) as it receives measurements z_k, $k = 1, 2, \ldots$ from a single observer (hence we drop index i_k from measurement notation). The standard deviation of range measurements depends on the distance between the observer located at $\mathbf{u}_k$ and detected object r at $\mathbf{x}_r$ as follows: $\sigma_k(\mathbf{x}_r) = 10 + 4 \cdot 10^{-5} \| \mathbf{x}_r - \mathbf{u}_k \|^2$.

5.2 Algorithm description

The main steps of the estimation algorithm with sensor control, including performance evaluation, are given in Algorithm 1. A detailed explanation of the main steps follows below.

Algorithm 1

1: **Input**:
2: - Sensor initial location
3: - Surveillance area $\mathcal{A}$
4: **for** $k := 1, \ldots K$ **do**
5: Get measurement set Z_k
6: **if** $k = 1$ **then**
7: Initialise particles $\mathbf{X}_1^i$, $i = 1, \ldots, N$
8: **end if**
9: Compute particle weights w_k^i
10: Resample & regularise particles
11: Compute point estimate $\hat{\mathbf{X}}_k$
12: Compute error metric $\bar{d}_k$ between $\hat{\mathbf{X}}_k$ and $\mathbf{X}$
13: Apply sensor control $\mathbf{u}_k$
14: **end for**
15: **Output**:
16: - Performance metric OSPA over time: $\bar{d}_{1:K}$

The particle filter (PF) requires that the maximum number of objects that can be present in $\mathcal{A}$, denoted $n_{\max}$, is specified a priori. Initialisation of particles (line 7 of Alg. 1) is carried out after receiving the first measurement set $Z_1 = \{z_{1,1}, \ldots, z_{1,m_1}\}$, of cardinality $m_1 = |Z_1|$. It consist of forming all possible subsets of possible multi-object RFSs (e.g. empty set, singletons for single object present, etc). For each subset, the objects are placed inside $\mathcal{A}$ along the arcs that correspond to range-measurements. The total number of particles is $N = 750$ and remains constant during the observation interval.

The computation of weights in line 9 of Alg. 1 is one of the critical steps both for the speed of execution and the accuracy of the PF. First note that since the multi-object state is static, we can effectively write the multi-object transitional pdf in (3) as $f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k) = \delta_{\mathbf{X}_{k+1}}(\mathbf{X}_k)$ which then leads to $f_{k+1|k}(\mathbf{X}_{k+1}|Z_{1:k}) = f_k(\mathbf{X}_k|Z_{1:k})$. In other words, there is no need for the prediction step in the multi-object Bayes recursions. Next, since we perform resampling (line 10 in Alg. 1) in every step of the PF, the unnormalised weights are computed as $\tilde{w}_k^i = g_k(Z_k|\mathbf{X}_k^i, \mathbf{u}_k)$, where the multi-object likelihood function is given by [15, p.410]:

$$g_k(Z_k|\mathbf{X}_k^i, \mathbf{u}_k) = f_c(Z_k)(1 - p_D)^{n_k^i} \times \sum_{\theta} \prod_{j:\theta(j)>0} \frac{p_D \, g_k(z_{\theta(j)}|\mathbf{x}_j^i, \mathbf{u}_k)}{(1 - p_D) \, \lambda \, c(z_{\theta(j)})} \tag{20}$$

Here $n_k^i = |X_k^i|$ represents the cardinality of particle i, and $\mathbf{x}_j^i \in \mathbf{X}_k^i$, for $j = 1, \ldots, n_k^i$. The sum on the RHS of (20) is over all possible associations θ between the particle set $\mathbf{X}_k^i$ and the measurement set Z_k, i.e. $\theta : \{1, \ldots, n_k^i\} \to \{0, 1, \ldots, m_k\}$. The term $f_c(Z_k)$ is the FISST pdf of clutter given by [15]

$$f_c(Z) = e^{-\lambda} \prod_{z \in Z} \lambda \, c(z).$$

In our case we set $c(z) = 1/R_{\max}$. Finally the single-object likelihood function $g_k(z|\mathbf{x})$ which features on the RHS of (20) follows from the measurement equation (19) and can be written as:

$$g_k(z|\mathbf{x}, \mathbf{u}_k) = \mathcal{N}(z; h_k(\mathbf{x}), \sigma_k^2(\mathbf{x})) \tag{21}$$

where $h_k(\mathbf{x}) = \| \mathbf{x} - \mathbf{u}_k \| = \sqrt{(x - \chi_k)^2 + (y - \varphi_k)^2}$ and $\sigma_k(\mathbf{x})$ was defined at the end of Sec. 5.1. Finally, once the unnormalised weights are computed, it is necessary to normalise them in the usual manner: $w_k^i = \tilde{w}_k^i / \sum_{i=1}^N \tilde{w}_k^i$.

The role of particle resampling (line 10 of Alg. 1) is to eliminate the samples with low importance weights and to clone the samples with high importance weights. After resampling, the weights of all particles become equal to $1/N$. Resampling can be done using any of the standard PF resampling methods, such as the stratified sampling [3], residual sampling [12], systematic resampling [9]. Our implementation is based on the systematic resampling, following the pseudo-code given in [19, Table 3.2]. After resampling it is useful to increase the particle diversity, which in this case was carried out by particle regularisation both in the state space and in the cardinality domain. Regularisation in the state space involves jittering the samples by a small random vector drawn from a Gaussian kernel of a bandwidth proportional to

the sample covariance matrix [17]. Regularisation in cardinality means that we force in a probabilistic manner some particles to modify their cardinality n_k^i as follows. A particle with $n_k^i < n_{\max}$ is forced, with probability $p_b \ll 1$ (birth), to add one more object to its set $\mathbf{X}_k^i$ (hence n_{k+1}^i becomes $n_k^i + 1$ with probability p_b), with the added object drawn uniformly from $\mathcal{A}$. Similarly, a particle with $n_k^i > 0$ is forced, with probability $p_d \ll 1$ (death), to remove one object (chosen randomly) from its set X_k^i (hence n_{k+1}^i becomes $n_k^i - 1$ with probability p_d). In our implementation we set $p_b = p_d = 0.025$.

The computation of the point estimate $\hat{\mathbf{X}}_k$ (line 11) involves the estimation of cardinality $\hat{n}_k$ and individual object states $\hat{\mathbf{x}}_{k,1}, \ldots, \hat{\mathbf{x}}_{k,\hat{n}_k}$. Cardinality is estimated using the MAP principle as:

$$\hat{n}_k = \arg \max_{0 \le j \le n_{\max}} \left(\sum_{i=1}^{N} \Delta[j, n_k^i] \right) \tag{22}$$

where $\Delta[i,j] = 1$ if $i = j$ and zero otherwise, is the Kroneker delta function. Due to the permutation symmetry of multi-object particles, individual object states ideally should be estimated by clustering the particles and finding the mean values of each individual cluster [11]. However, as explained in [2], due to resampling, the particles have a self-resolving property. For this reason the individual object states in our simulations are obtained by simply averaging the particles characterised by cardinality estimate $\hat{n}_k$ obtained from (22).

In order to control the observer we need first to define the set of admissible actions $\mathbb{U}_k$, from which the sensor control $\mathbf{u}_k$ is selected, see line 13 of Alg. 1. If the current position of the observer is $\mathbf{v}_k = [\chi_k \;\; \varphi_k]^T$, its one-step ahead admissible locations are adopted as follows:

$$\mathbb{U}_k = \big\{ \, (\chi_k + j\rho_0 \cos(\ell\theta_0), \varphi_k + j\rho_0 \sin(\ell\theta_0)) \, ; \\ j = 0, \ldots, N_\rho; \ell = 1, \ldots, N_\theta \big\} \tag{23}$$

where $\theta_0 = 2\pi/N_\theta$ and ρ_0 is a conveniently selected radial step size. In this way the observer can stay in its current position ($j = 0$) or move radially in incremental steps. In our study we adopted $N_\rho = 2$, $N_\theta = 8$ and $\rho_0 = 50$m. We thus have 17 control options, and for each we compute the reward function $\mathbb{E}[\mathcal{D}]$ according to (18). If a control vector $\mathbf{u}_k \in \mathbb{U}_k$ is outside $\mathcal{A}$, its reward is set to $-\infty$; in this way the observer is always kept inside the surveillance area. In the practical implementation of (18) we generated $M = 200$ "future" measurement sets $\mathbf{Z}_{k+1}^j$, $j = 1, \ldots, M$. Since both clutter and probability of detection are homogeneous, these measurement sets are created in the "ideal" manner, that is with no clutter and with $p_D = 1$ (a similar argument was made in [16]). The choice of α in (18) is based on the earlier observation [7] that when two densities are similar (as they are in our case), the choice of $\alpha = 0.5$ provides the maximum discrimination between them.

5.3 Performance evaluation

The error performance of the proposed algorithm (the error between the estimated and true multi-target states, $\hat{\mathbf{X}}_k$ and $\mathbf{X}$, respectively) is measured (line 12 of Alg. 1) using the optimal sub-pattern assignment (OSPA) metric or multi-target miss-distance proposed in [20]. We use the OSPA metric because it jointly captures the differences in both the cardinality and the individual elements between two finite sets, in a mathematically consistent yet intuitively meaningful way. Let $d_c(\mathbf{x}, \mathbf{y}) := \min(c, \|\mathbf{x} - \mathbf{y}\|)$ for $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, and Π_k denote the set of permutations on $\{1, 2, \ldots, k\}$ for any $k \in \mathbb{Z}^k$. Then, for $p \ge 1$, $c > 0$, and $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, the OSPA is defined for three separate cases. If $m = n = 0$, then $\bar{d}_{c,p}(\mathbf{X}, \mathbf{Y}) = \bar{d}_{c,p}(\mathbf{Y}, \mathbf{X}) = 0$. Otherwise, if $m \le n$:

$$\bar{d}_{c,p}(\mathbf{X}, \mathbf{Y}) \triangleq \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + c^p(n-m) \right) \right)^{\frac{1}{p}} \tag{24}$$

and if $m > n$, $\bar{d}_{c,p}(\mathbf{X}, \mathbf{Y}) := \bar{d}_{c,p}(\mathbf{Y}, \mathbf{X})$. The two parameters of OSPA metric are order p and the cut-off c; the order p determines the sensitivity to outliers, while the cut-off c determines the relative weighting of the penalties assigned to cardinality and localization errors. We adopt $p = 2$ and $c = 100$m in this study.

When we present numerical results in the next section, we will show not only the OSPA values, but also the cardinality and localisation errors, separately. These errors are defined as [20]:

$$e_{c,p}^{\text{card}}(\mathbf{X}, \mathbf{Y}) = \left(\frac{c^p(n-m)}{n} \right)^{\frac{1}{p}} \tag{25}$$

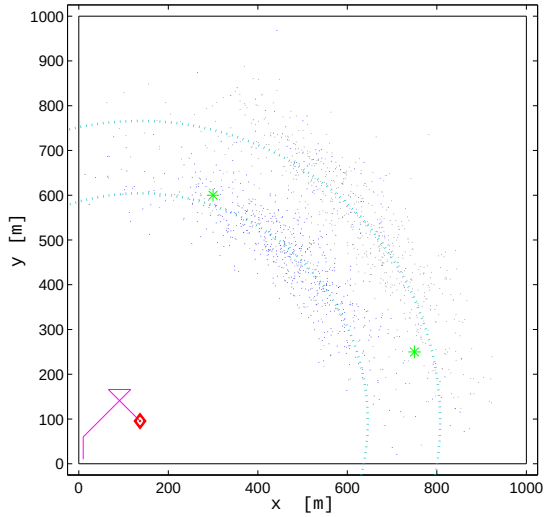
$$e_{c,p}^{\text{loc}}(\mathbf{X}, \mathbf{Y}) = \left(\frac{1}{n} \cdot \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p \right)^{\frac{1}{p}} \tag{26}$$

if $m \le n$, and $e_{c,p}^{\text{card}}(\mathbf{X}, \mathbf{Y}) := e_{c,p}^{\text{card}}(\mathbf{Y}, \mathbf{X})$ and $e_{c,p}^{\text{loc}}(\mathbf{X}, \mathbf{Y}) := e_{c,p}^{\text{loc}}(\mathbf{Y}, \mathbf{X})$ if $m > n$.

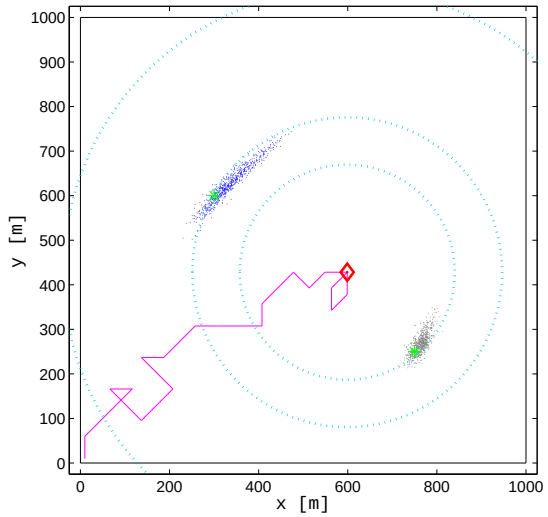
5.4 Numerical results

Fig.1 shows a single run of the proposed algorithm after (a) $k = 7$ and (b) $k = 24$ time steps. The pink line indicates the observer path; the current observer location is denoted by a diamond. The black and blue dots indicate the particles $\mathbf{X}_k^i$ characterised by cardinality $n_k^i = 2$. The cyan arcs/circles correspond to range measurements at time k. The true object locations are indicated with green asterisks. The probability $Pr\{n = 2\}$

equals 0.93 and 0.95 after $k = 7$ and $k = 24$ time steps. In a typical run of the algorithm, the sensor is controlled at first towards the mid point between the sources, as illustrated in Fig.1. Afterwards the algorithm tends to favour observer paths which approach and circle around one of the objects, followed by a similar motion pattern around the other object. The OSPA metric versus time for the run considered in Fig.1 is shown in Fig.2.



(a)



(b)

Fig. 1. Localisation with sensor control: (a) after $k = 7$ time steps; (b) after $k = 24$ time steps. The pink line indicates the observer path; the current observer location is denoted by a diamond. The black and blue dots indicate the particles $\mathbf{X}_k^i$ characterised by cardinality $n_k^i = 2$. The cyan arcs/circles correspond to range measurements. The true object locations are indicated by green asterisks. Note the self-resolution of permutation symmetry.

Finally we compare the performance of the proposed sensor management scheme against three alternatives. All three are identical to the described algorithm, except

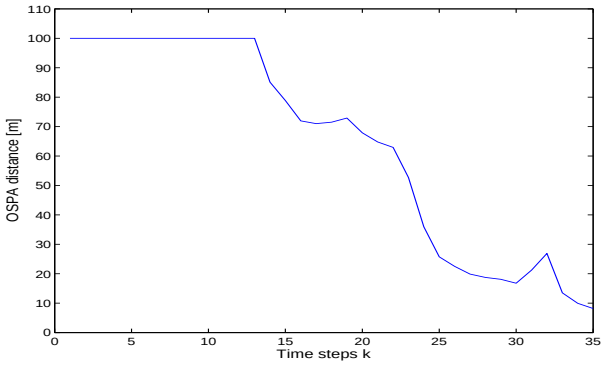


Fig. 2. OSPA metric versus time for the run considered in Fig.1

in the choice of the sensor control vector at time k from the admissible set $\mathbb{U}_k$. The three alternative schemes are described below.

(1) Uniformly random scheme. In this scheme, we choose $\mathbf{u}_k$ randomly from $\mathbb{U}_k$.

(2) The approximate Fisher information gain. Suppose the estimate of the multi-object state (line 11 in Alg. 1) is $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_{k,1}, \ldots, \hat{\mathbf{x}}_{k,\hat{n}_k}\}$. By taking action $\mathbf{u}_k$ and assuming an ideal detection ($p_D = 1$ and no false detections), let the measurement set be denoted by $Z_k = \{z_{k,1}, \ldots, z_{k,\hat{n}_k}\}$, with $z_{k,m} = h_k(\hat{\mathbf{x}}_{k,m}) = ||\hat{\mathbf{x}}_{k,m} - \mathbf{u}_k||$. Assuming furthermore a significant separation of objects in the state space $\mathcal{X}$, one can approximate the multi-object likelihood function (20) by:

$$g_k(Z_k|\hat{\mathbf{X}}_k, \mathbf{u}_k) \approx \prod_{m=1}^{\hat{n}_k} g_k(z_{k,m}|\hat{\mathbf{x}}_{k,m}, \mathbf{u}_k) \tag{27}$$

where the single-object likelihood $g_k(z|\mathbf{x}, \mathbf{u}_k)$ was given by (21). The Fisher information gain is then given by:

$$\mathbf{J}(\mathbf{u}_k) \triangleq -\mathbb{E}\left[\nabla\nabla^T \log g_k(Z_k|\hat{\mathbf{X}}_k, \mathbf{u}_k)\right] \tag{28}$$

$$\approx \sum_{m=1}^{\hat{n}_k} \frac{1}{\sigma_k^2(\hat{\mathbf{x}}_{k,m})} \mathbf{H}_{k,m}^T \mathbf{H}_{k,m} \tag{29}$$

where $\mathbf{H}_{k,m}$ is the Jacobian of $h_k(\hat{\mathbf{x}}_{k,m})$. The reward function in this scheme is the trace of matrix $\mathbf{J}(\mathbf{u}_k)$.

(3) PENT. The PENT criterion was introduces for the control of sensors with a finite field-of-view (FoV), for the purpose of selecting the action which will maximize the number of objects to be seen by the sensor. Again let $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_{k,1}, \ldots, \hat{\mathbf{x}}_{k,\hat{n}_k}\}$. By taking action $\mathbf{u}_k$, an ideal noise and clutter free measurement set with $p_D = 1$ is then: $\{z_{k,1}, \ldots, z_{k,\hat{n}_k}\}$, with $z_{k,m} = h_k(\hat{\mathbf{x}}_{k,m}) = ||\hat{\mathbf{x}}_{k,m} - \mathbf{u}_k||$. The PENT [16], as a reward function in our case

study, can then be written as:

$$\hat{\nu}_{k+1}(\mathbf{u}_k) = C + p_D \sum_{m=1}^{\hat{n}_k} \left(1 - \frac{\lambda\, c(z_{k,m})}{\lambda\, c(z_{k,m}) + p_D \mathcal{I}_k(\mathbf{u}_k)}\right) \tag{30}$$

where C is a constant independent of action $\mathbf{u}_k$ and

$$\mathcal{I}_k(\mathbf{u}_k) = \int g_k(z_{k,m}|\mathbf{x}, \mathbf{u}_k)\, D_k(\mathbf{x}|Z_{1:k}, \mathbf{u}_{0:k-1}) d\mathbf{x}, \tag{31}$$

with $D(\mathbf{x})$ being the intensity function (a.k.a. the probability hypothesis density) of RFS $\mathbf{X}$. The intensity function can be computed from the FISST density $f(\mathbf{X})$ as $D(\mathbf{x}) = \int \delta_{\mathbf{X}}(\mathbf{x}) f(\mathbf{X}) \delta\mathbf{X}$. In our case the FISST density is the posterior $f_k(\mathbf{X}_k|Z_{1:k}, \mathbf{u}_{0:k-1})$ which is approximated by particles as $\frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{X}_k^i}(\mathbf{X}_k)$. It can be shown that the particle representation of the posterior intensity function (after resampling) is given by:

$$D_k(\mathbf{x}|Z_{1:k}, \mathbf{u}_{0:k-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{\mathbf{w}\in\mathbf{X}_k^i} \delta(\mathbf{x} - \mathbf{w}) \tag{32}$$

$$= \frac{1}{N} \sum_{j=1}^{N^*} \delta(\mathbf{x} - \mathbf{x}_k^j) \tag{33}$$

where $\{\mathbf{x}_k^j\}_{j=1}^{N^*}$ is the set of all individual objects represented by multi-object particles $\{\mathbf{X}_k^i\}_{i=1}^{N}$. Using (33), the integral in the denominator of (30) is approximated as:

$$\mathcal{I}_k(\mathbf{u}_k) \approx \frac{1}{N} \sum_{j=1}^{N^*} g_k(z_{k,m}|\mathbf{x}_k^j, \mathbf{u}_k). \tag{34}$$

The OSPA metrics, as well as the cardinality and localisation errors, of all four sensor management approaches are averaged over 200 independent Monte Carlo runs, with results shown in Fig.3. First we note from Fig.3 that the main contributor to OSPA in this case study is the localisation error (cardinality errors are comparatively much smaller). Second, the proposed alpha-divergence information driven sensor control clearly performs the best: its OSPA distance reduces over time to the lowest value. Somewhat surprisingly, random action selection performs quite well and better than Fisher-information and the PENT. The Fisher information reduces OSPA quickly at initial stages (between the time step $k = 7$ and $k = 18$ it performs the best), but subsequently it does not improve significantly. The PENT performs very poorly, but this is not surprising: first, the PENT is not concerned with localisation uncertainty (which clearly dominates in our case study, due to range-only measurements); second, since all objects are seen by the sensor irrespective of the action, there is nothing for PENT to maximize.

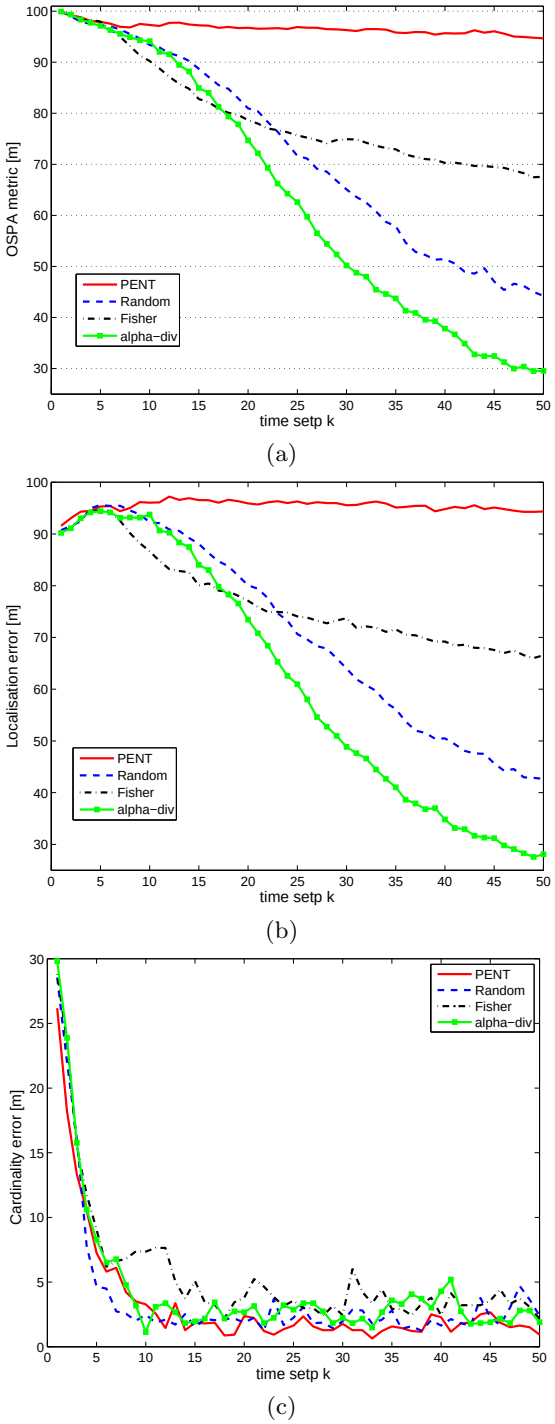


Fig. 3. Error performance of the four sensor management schemes, averaged over 200 Monte Carlo runs: (a) OSPA metrics; (b) Localisation error; (c) Cardinality error

6 Conclusions

The paper presented an information based sensor management scheme for Bayesian multi-object nonlinear filtering in the framework of random finite sets. The

reward function is computed using the Rényi alpha divergence between the multi-object prior and the multi-object posterior probability densities. The proposed sensor management algorithm is straightforward to compute for sequential Monte Carlo implementations of the Bayesian multi-object filter. As a proof of concept, the benefits of the sensor management scheme are illustrated in a multi-object localisation study, where range-to-object noisy measurements, affected by unreliable detection, are available for estimation. The proposed sensor management scheme in this case study clearly outperforms the alternatives. Future work will look at simpler and computationally improved approximations of the reward function, specifically designed for simplified multi-object Bayes filters, such as the PHD filter and the multi-Bernoulli filter [15,22].

References

[1] D. P. Bertsekas and D. A. Castanon. Rollout algorithms for stochastic scheduling problems. *Jour. Heuristic*, 5:89–108, 1999.

[2] Y. Boers and H. Driessen. The mixed labeling problem in multi target particle filtering. In *Proc. 10th Int. Conf. Information Fusion*, Québec, Canada, July 2007. ISIF.

[3] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for non-linear problems. *IEE Proc. Part F: Radar and Sonar Navigation*, 146(1):2–7, Feb 1999.

[4] D. A. Castanón and L. Carin. Stochastic control theory for sensor management. In A. O. Hero III, D. A. Castanón, D. Cochran, and K. Kastella, editors, *Foundations and Applications of Sensor Management*, chapter 2, pages 7–32. Springler, 2008.

[5] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[6] A. El-Fallah, A. Zatezalo, R. Mahler, R. Mehra, and D. Donatelli. Dynamic sensor management of dispersed and disparate sensors for tracking resident space objects. In *Proc. SPIE*, volume 6968, 2008.

[7] A. O. Hero, C. M. Kreucher, and D. Blatt. Information theoretic approaches to sensor management. In A. O. Hero, D. Castanòn, D. Cochran, and K. Kastella, editors, *Foundations and applications of sensor management*, chapter 3, pages 33–57. Springer, 2008.

[8] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[9] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian non-linear state space models. *Journal Of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[10] C. M. Kreucher, A. O. Hero, K. D. Kastella, and M. R. Morelande. An information based approach to sensor management in large dynamic networks. *Proc. of the IEEE*, 95(5):978–999, May 2007.

[11] C. M. Kreucher, M. Morelande, K. Kastella, and A. O. Hero III. Joint multi-target particle filtering. In A. O. Hero, D. Castanòn, D. Cochran, and K. Kastella, editors, *Foundations and applications of sensor management*, chapter 4, pages 59–93. Springer, 2008.

[12] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamical systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[13] R. Mahler. Global posterior densities for sensor management. In *Proc. SPIE*, volume 3365, pages 252–263, 1998.

[14] R. Mahler. Multitarget sensor management of dispersed mobile sensors. In D. Grundel, R. Murphey, and P. Pardalos, editors, *Theory and algorithms for cooperative systems*, chapter 12, pages 239–310. World Scientific books, 2004.

[15] R. Mahler. *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.

[16] R. Mahler and T. Zajic. Probabilistic objective functions for sensor management. In *Proc. SPIE*, volume 5429, pages 233–244, 2004.

[17] C. Musso, N. Oudjane, and F. LeGland. Improving regularised particle filters. In A. Doucet, N. deFreitas, and N. J. Gordon, editors, *Sequential Monte Carlo methods in Practice*. Springer-Verlag, New York, 2001.

[18] D. Pollard. *A user's guide to measure theoretic probability*. Cambridge University Press, 2001.

[19] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.

[20] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Processing*, 56(8):3447–3457, Aug. 2008.

[21] B. N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for multi-target filtering with random finite sets. *IEEE Trans. Aerospace & Electronic Systems*, 41(4):1224–1245, Oct. 2005.

[22] B.-T. Vo, B.N. Vo, and A. Cantoni. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Processing*, 57(2):409–423, 2009.

[23] J. Witkoskie, W. Kukunski, S. Theophanis, and M. Otero. Random set tracker experiment on a road constrained network with resource management. In *Proc. 9th Int. Conf. Information Fusion*, Florence, Italy, 2006.

[24] A. Zatezalo, A. El-Fallah, R. Mahler, R. K. Mehra, and K. Pham. Joint search and sensor management for geosynchronous satellites. In *Proc. SPIE*, volume 6968, 2008.