Target Tracking using Proximity Binary Sensors

Qiang Le
Department of Engineering
Hampton University
Hampton, VA 23668
qiang.le@hamptonu.edu

Lance M. Kaplan
US Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197
lkaplan@ieee.org

Abstract—This paper investigates the feasibility of a mesh network of proximity sensors to track multiple targets. In such a network, the sensors report a detection when a target is within the proximity; otherwise, the sensors report no detection. Previous work has revealed the potential of target localization and tracking for a single target using these binary reports. This work introduces a particle-based probability hypothesis density (PHD) filter that is able to track multiple targets using the binary reports from a proximity sensor network. Furthermore, this work modifies another particle-based multitarget tracker for proximity sensors, namely the ClusterTrack, from 1-D tracking to 2-D. The simulations demonstrate that the PHD is able to outperform the ClusterTrack in terms of both accuracy of localization and estimating the number of targets.

Table of Contents

1. Introduction

Wireless sensor networks have prove useful for numerous activities including environmental monitoring, structural health monitoring, and surveillance [1]. The main idea behind sensor networks is that many cheap sensors working together can achieve similar if not better performance than a more expensive monolithic senor system due the exploitation of spatial diversity. Among the cheapest and most energy efficient devices are proximity (or binary) sensors. These are low power devices that report a detection whenever a target is nearby. Typically, these sensors are used to cue more sophisticated senors that require more power in order to classify and track the targets. This paper investigates the feasibility of a mesh network of proximity sensors to track multiple targets. In such a network, the no-detection report is as valuable as a detection report. Previous work has revealed the potential of target localization and tracking for a single target [2], [3], [4], [5]. There are two models to describe the behavior of the proximity sensor that drive localization algorithms. Many approaches are based on the disc model that assume a sensor detects a target when and only when the target's distance to the sensor is below a sensing radius [4], [5]. The disc model ignores physical facts such as the additive effects of the energy radiating from multiple target will extend the sensing range. Other approaches uses a probabilistic model for the sensors that is based on a physical energy loss as the energy is radiated from each of the targets [2], [3]. For multiple-target tracking, Singh, *et al.*, proposed *ClusterTrack*, which is a clustering-based particle filter approach to track targets based on the disk model [6]. However, the tracking has only been demonstrated in 1-D space. In our previous work, we presented a multiple target maximum likelihood (ML) localization algorithm based on the probabilistic model when the number of targets is known [7]. In more recent work, we have expanded the work to encompass tracking when the number of the targets is unknown by exploiting a particle-based implementation of the probability hypothesis density (PHD) filter [8]. Specifically, we modified the likelihood function in the PHD filter originally proposed by [9] to account for the fact that the binary measurements are intertwined with all sensors in the scene. This work compares the PHD against ClusterTrack for both 1-D and 2-D scenarios.

The applications of the PHD filter have been seen in sonar image, visual tracking, and radar tracking [10], [11], [12], [13], [14], [15]. For example, [10], [11] implemented the PHD filter to identify the underwater obstacles that would need to be avoided in navigation using forward-scan sonar images. [12] used the PHD filter to track a random number of pedestrians in image sequences and derive their location sequences. [13] applied the PHD filter to target tracking using both range and Doppler measurements. [14], [15] provided a more generic PHD filter that incorporates Cartesian target position measurements directly or through bearing and range measurements. To the best of our knowledge, the PHD filter has not yet been used in proximity-based multiple target tracking.

IEEEAC Paper#1802, Version 2 Updated 02/02/2011.

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0041. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

These PHD filters assume that the measurements are independent and associated to one target. As a result, the measurement update of the PHD has a simple analytical form. In general, the PHD filter is a sequential Bayesian tracker that is only following the evolution of the first order multitarget moment of the multitarget density. A motivation for the PHD is that following the multitarget density itself is too cumbersome for practical implementation, e.g., via particle filtering. In this paper, we apply PHD filtering for a mesh network of proximity senors where the measurements do not directly associate to a single target. As is shown in the next section, the response of the proximity sensor depends on its location relative to the location of all targets within the scene. As a result, the measurements, i.e., the binary outputs of the proximity sensors, are intertwined with all targets. Therefore, tracking with binary sensors requires a modification of the likelihood (measurement) update equation in the original PHD formulation. In our recent paper [8], we derived the modified measurement update equation by using the same Bayesian update method as in [9], with the exception that the multisensor/target likelihood is not separable. The likelihood is based on the probabilistic model. In this paper, we also investigate an alternative likelihood update that is based on the disc model.

The ClusterTrack particle filter was first proposed for the disc sensor model and tested on 1-D data [6]. The particle at a given snapshot time t does not simply represent a possible target position at time t, but a target trajectory till time t. The hypothesized target positions in a particle are random samples over a feasible target area (FTA) derived from the sensor reports. The FTA samples are integrated over time to form particles. The particles are ranked to favor smoother trajectories and clustered so to help ensure that particles retained for the next measurement cycle are not dominated by a single track, i.e., target. Whether or not the retained particles can catch all the target paths depends on a number of factors such as how the FTA samples are obtained, how the particles are ranked, and how they are clustered. The ClusterTrack as proposed in [6] was only tested for 1-D tracking scenarios. In fact, the obvious extension to 2-D tracking is problematic. In this paper, we consider multiple extensions of ClusterTrack to 2-D and compare against the PHD filter.

The paper is as organized as follows. Section 2 describes the two proximity sensor models. Then, Section 3 reviews the particle filter implementation of the ClusterTrack method for the proximity sensor network and discusses modifications for 2-D tracking. Section 4 presents the particle filter implementation of the PHD method for proximity sensors. Simulations are provided in Section 5 to compare the performance of the PHD filter and ClusterTrack method. Finally, Section 6 provides concluding remarks.

2. SENSOR MODEL

As discussed in the previous section, methods to localize and track targets are based primarily on two different sensing models. The following subsections review each model.

Probabilistic Model

The probabilistic model accounts for the effects of sensor noise so that a sensor will report (or not report) a detection based on some probability of detection. This detection probability goes up as the signal from the targets increases, which happens as targets move closer to the sensor. The probabilistic model used in this paper is taken from [3]. Specifically, the received power p_i^t, i.e., the power measurement of the i-th sensor at time t, is given by

$$p_i^t = \sum_{k=1}^{N} p_{0,k}^t \left(\frac{r_{0,k}}{r_{i,k}^t} \right)^{\alpha} + v_i^t, \tag{1}$$

where $p_{0,k}^t$ is the power measured at a reference distance $r_{0,k}$ for the k-th target, N is the total number of targets, $r_{i,k}^t$ is the relative distance between the i-th sensor and the k-th target, α is the attenuation parameter that depends on the transmission medium, and

$$v_i^t \sim \mathbf{N}\left(\mu_v, \sigma_v^2\right).$$

The mean and variance of the error v_i^t is derived from the zero mean measurement noise of variance σ^2 for the case that the measured power is the result of integrating the square of the measurements over L samples. This model assumes that the target signals are decorrelated so that the powers due to the target signals sum up incoherently. As a result of integrating over L samples, $\mu_v = \sigma^2$ and $\sigma_v^2 = 2\sigma^4/L$. This paper assumes the reference distances $r_{0,k} = r_0$ and target powers $p_{0,k} = p_0$ for all K targets, and r_0 and p_0 are known.

The assumption that the reference power for each target $p_{0,k}$ is equivalent is, in general, restrictive. However, if the reference powers for the targets are completely arbitrary, then the multiple target localization problem is ill posed. For instance, it would be ambiguous if the detection pattern of the senors is caused multiple closely spaced low reference power targets or a single high reference power target. In many applications it is reasonable to assume that the reference power is unknown within a bounded range of values. As an initial step, we assume that the spread between the upper and lower bounds is zero, i.e., $p_{0,k}$ is constant. Future work can investigate the effects of the size of the spread to the ability to estimate the number of targets and their respective locations.

The i-th sensor measures the received power p_i^t, processes it locally, and reports a single binary digit: '1' for the presence of one or more targets or '0' for the absence of any target. The decision follows the rule

$$d_i^t = \begin{cases} 1 & p_i^t > \lambda, \\ 0 & p_i^t \leq \lambda. \end{cases} \tag{2}$$

The probability of false alarm P_{fa} is the probability that d_i^t exceeds the threshold when $p_{0,k}^t = 0$. For desired value of P_{fa}, the threshold λ is computed as:

$$\lambda = \sigma_v Q^{-1}(P_{fa}) + \mu_v, \tag{3}$$

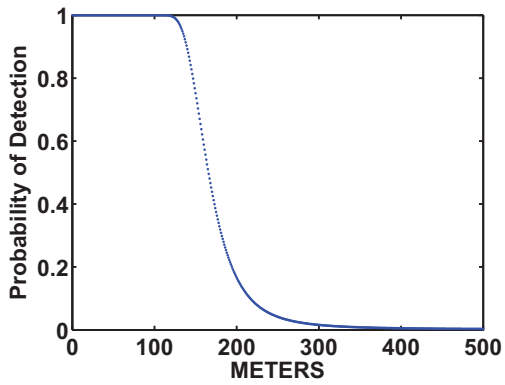


Figure 1. P_d vs. the distance, where P_{fa}=0.001, $\alpha = 2$, $\sigma = 0.5$, $p_0 = 3000$, $r_0 = 1$m, and $L = 100$.

where $Q(\cdot)$ is the Q-function

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt.$$

Furthermore, the probability of detection P_d at the i-th sensor, for a given threshold (or P_{fa}), as a function of the distances to the k targets, is

$$\begin{aligned} P_d(r_i^t) &= \text{Prob}(d_i^t = 1|\text{targets}), \\ &= \text{Prob}(p_i^t > \lambda|\text{targets}), \\ &= Q\left(\frac{\lambda - \sum_{k=1}^K p_{0,k}^t \left(\frac{r_{0,k}^t}{r_{i,k}^t}\right)^\alpha - \mu_v}{\sigma_v} \right), \end{aligned} \quad (4)$$

where $r_i^t = \left[r_{i,0}^t, \ \ldots, \ r_{i,k}^t \right]$. Fig. 1 illustrates P_d as a function of the distance of a single target to a sensor. The figure demonstrates that P_d as given by (4) behaves as a sigmoid function that steps down from a value near one to a value near P_{fa} over a transition region of about 200m.

For multiple targets, the incoherent addition of the target powers does extend the sensing range of the sensors. This fact actually allows for one to infer the number of targets when looking at the distribution of detection reports in a mesh network of proximity sensors. If the targets are far apart, one would expect to see sperate clusters of sensors reporting detections around each target. When the targets are near each other, only one cluster emerges. The size of the cluster increases as more targets are in the same vicinity due to the incoherent addition of power in (4), and it is the size of the cluster that provides the clue to the number of targets. The ability to estimate the number of targets depends on having a good handle on the target radiation power $p_{0,k}$. If $p_{0,k}$ is not known, then the number of targets within the cluster of sensing nodes is ambiguous, i.e., is there one loud target or multiple quiet targets? For this reason, we assume in this paper that $p_{0,k} = p_0$. Future work will investigate robustness of this assumption when the targets radiation power is known to fall within a certain range.

Disc Sensor Model

Many papers exploiting proximity sensors assume the *disc* sensor model, where a sensor always detects one or more targets if they are within the detection range r_{eff} and never detects any target if all are further than r_{eff} from the sensor:

$$P_d(r_i^t) = \begin{cases} 1 & \exists k, r_{i,k}^t \le r_{eff}, \\ 0 & \text{Otherwise.} \end{cases} \quad (5)$$

The disc model ignores the physical process as discussed in previous subsection that enables sensors to detect the targets. For instance, the presence of multiple targets extends the detection range of the sensor due to the superposition of the target energy. As discussed, it is the superposition of power property that allows one to estimate the the number of targets. For the disc model, the number of targets is always ambiguous. Consider a proximity sensor network with one cluster of detections. The cluster could be the result of one target or hundreds of targets within the same vicinity. Beyond the fact that the disc model is an idealized representation of the behavior of the sensor, the inability of the disc sensing model to disambiguate the number of targets is a severe weakness that can affect the performance of tracking algorithms based on this model as shown in Section 5.

3. CLUSTERTRACK METHOD

Based upon the sensor readings, let $F(t)$ the feasible target area (FTA) at time t computed by:

$$F = \bigcup_{i \in I} D_i - \bigcup_{j \in Z} D_j, \quad (6)$$

where I is the set of sensors whose binary output is 1, Z is the set of sensors whose binary output is 0, and D_i is the sensing disc of the sensor i.

Let us denote the kth particle at time t by P_t^k for $1 < k < K$. For each k, P_t^k is a vector of length t: $(x_k(1), \cdots, x_k(t))$ where $x_k(t)$ is the 2-D target position in the k-th particle. Hence each particle is an estimate of a target trajectory till time t.

The ClusterTrack method is designed to present a single target from dominating all particles. Instead of looking for clusters at the end, it clusters the particles throughout the tracking process. It retains a maximum of K particles at time instant t, that is, $K_t \le K$. Further, the chosen particles are subject to the constraint that the number of particles per cluster does not exceed a threshold H. The cost for the kth particle P_t^k at time t is defined as

$$\sum_{a=2}^{t-1} c(a) \quad (7)$$

where $c(t)$ as defined in [6] is the norm of the change in velocity, i.e.,

$$c(t) = ||x(t+1) - x(t) - (x(t) - x(t-1))||. \quad (8)$$

For 1-D tracking as demonstrated in [6], this cost function can be effective. However, for 2-D tracking, the cost function seems to ignore abrupt changes in headings (which can occur at small velocities). Thus, we extend ClustTrack for 2-D by also considering an alternative cost function that represents the norm of the change in direction, i.e.,

$$c(t) = |\theta(t) - \theta(t-1)| \text{ where } \theta(t) = \arctan\left(\frac{x(t)_y}{x(t)_x}\right). \quad (9)$$

Given K_{t-1} particles $\{P_{t-1}^k\}_{k=1}^{K_{t-1}}$ and m new points randomly sampled over the FTA $F(t)$ at time t, the particle filter proceeds via the following steps:

1. Extend the set $\{P_{t-1}^k\}_{k=1}^{K_{t-1}}$ to time t, by generating a total of mK_{t-1} particles.
2. Sort the mK_{t-1} particles in ascending order of cost to obtain the set $\{\hat{P}_1, \cdots, \hat{P}_{mK_{t-1}}\}$ where the cost of a particle is computed by (7).
3. Put $\hat{P}_1$ in Cluster$_1$
4. $P_t^1 := \hat{P}_1, N_c := 1;$ Count$_1 := 1; i := 2; k := 1.$
5. while ($i \leq L$ and $k \leq K$) do
    if ($\hat{P}_i \in$ Cluster$_j$ for some j), then
        if Cluster$_j < H$, then
            Count$_j :=$ Count$_j + 1$
            $k := k + 1$
            $P_t^k := \hat{P}_i$
        else
            Abandon $\hat{P}_i$
        end if
    else
        Make new cluster for $\hat{P}_i$
        $N_c := N_c + 1; k := k + 1; P_t^k := \hat{P}_i$
    end if
    $i := i + 1$
  end while
6. Let $t := t + 1$ and goto Step #1.

In the ClusterTrack method, Cluster$_j$ represents the jth cluster, count$_j$ denotes the number of particles retained in Cluster$_j$, N_c is the number of clusters, H is the maximum number of particles to be retained in particular cluster, and L is the maximum number of particles to be inspected.

The decision in Step 5 (whether the particle $\hat{P}_i$ belongs any of the existing clusters) is made as follows. We define a cluster head for each existing cluster, denoted by CH_j, as the first particle joining the jth cluster. The distance between two particles P_1 and P_2 of length t is denoted by $D(P_1, P_2)$, and is measured as the sum of the norm differences in the elements

$$D(P_1, P_2) = \sum_{l=1}^{t} \|P_2[l] - P_2[l]\|. \quad (10)$$

If the smallest distance $D(\hat{P}_i, CH_j)$ from the i-th particle to a cluster head is less than a threshold $D_0(t)$, then particle $\hat{P}_i$ belongs to the closest cluster head. Otherwise, ClusterTrack creates a new cluster head as particle $\hat{P}_i$. The threshold $D_0(t)$ determines how many cluster heads are formed. Some discussion on the choice of $D_0(t)$ appears in [6] and it was concluded that $D_0(t)$ should be chosen between $r_{eff}t$ and $2r_{eff}t$ for the cost given by (8) to ensure that "ClusterTrack catches all of the paths most of the time" [6]. On the other hand, the number of particles allowed in a cluster, H, also plays a critical role in determining the number of resulting clusters. Intuitively, the lower H is, the more clusters. Parameters such as $H, D_0(t), K$ and L together determine which clusters emerge. It is not clear how to balance these parameters to guarantee that all target paths are captured. Furthermore, it is not at all clear if the parameters can be chosen so that the number of cluster heads are related to the actual number of targets in the scene.

4. PHD

The multitarget density function incorporates the probability mass for n targets and the density for target states x_i for $i = 1, \ldots, n$, $f(n; x_1, \ldots, x_n)$ so that

$$\text{Prob}(\#\text{targets} = n) = \int f(n; x_1, \ldots, x_n) dx_1 \cdots dx_n. \quad (11)$$

In this paper, the target state x_i is a four element column vector representing the 2-D position and velocity of the target. Due to the curse of the dimensionality, the particle filter implementation of the multi-objective filter would require too many particles. The density can be summarized by its PHD, which represents the support of the density for a particular target state over all possible target count hypotheses, i.e.,

$$D(x) = \sum_{n=1}^{\infty} \int n f(n; x, x_1, \ldots, x_{n-1}) dx_1 \cdots dx_{n-1}. \quad (12)$$

Therefore, the PHD represents the density of the expected number of objects at a particular target state x; that is, the integral of the PHD in the target space represents the expected number of objects.

Given P particles to represent the posteriori PHD at $t-1$, i.e.,

$$D(x_{t-1}|t-1) \approx \sum_{p=1}^{P} w_{t-1}^{(p)} \delta(x_{t-1} - x_{t-1}^{(p)}), \quad (13)$$

the estimated number of targets is the integral of the PHD so that

$$\hat{N}_{t-1} \approx \sum_{p=1}^{P} w_{t-1}^{(p)}, \quad (14)$$

and the predicted PHD at t is obtained by diffusing the particles so that

$$D(x_t|t-1) \approx \sum_{p=1}^{P} w_{t-1}^{(p)} \delta(x_t - x_{t-1}^{*(p)}), \quad (15)$$

where $x_{t-1}^{*(p)} = Fx_{t-1}^{(p)} + Q^{\frac{1}{2}}v^{(p)}$, $v^{(p)}$ is sampled from a white Gaussian random number generator, and F and Q are the state transition and process noise covariance matrices, respectively. To perform the likelihood update, similar to [9], the PHD is expanded into a multitarget density by assuming a Poisson Point Process. Namely, $P_{ex} \geq P$ particles are created where the k-th particle represents $n^{(k)}$ targets, and $n^{(k)}$ is drawn from a Poisson distribution with mean $\hat{N}_{t-1}$ where $P_{ex} = Pn_{ex} + J$. Inside the P_{ex} multiple target particles, the J innovative particles are added to tackle the target maneuvering, target birth, and target death issues. The factor n_{ex} represents the expansion for number of particles to represent the multitarget density via the PHD. In our simulations, $n_{ex} = 1$. The $n^{(k)}$ target state vectors are drawn from the PHD particles, where $x_{t-1}^{*(p)}$ is selected with probability $w_{t-1}^{(p)}$. Due to the likelihood update, the weight associated to the k-th particle is a function of the senors' probability of detection,

$$w_k^f = \frac{1}{C}\prod_{m=1}^{N_s}\left(P_d(S_m, x_{t,1}^{(k)}, \ldots, x_{t,n^{(k)}}^{(k)})\right)^{d_{t,m}} \left(1 - P_d(S_m, x_{t,1}^{(k)}, \ldots, x_{t,n^{(k)}}^{(k)})\right)^{1-d_{t,m}}, \quad (16)$$

where C is a normalization constant so that the weights sum to one, and S_m and $d_{t,m}$ are the location and binary output of the m-th sensor at time t, respectively. When sensors follow the probabilistic sensor model, the weight update should follow (4). Otherwise when sensors follow the disc sensor model, the weight update should follow (5).

The size of each innovative particle $n^{(k)}$ is drawn from a Poisson distribution with mean $\hat{N}_{t-1}$. Each of the $n^{(k)}$ target states are random samples drawn from a proposal distribution $p(\cdot|Y_t)$ where Y_t represents the measurements, i.e., detection reports, at time t. Once these particles are drawn, they are also weighted via (16). In [8], the proposal distribution for the 2-D position elements is simply uniform over the whole surveillance field, and the 2-D velocity elements are zero. This proposal density can hypothesize particles with very low weights that do not survive a later resampling stage. Therefore, it is better to draw innovative samples in more likely locations to enhance their effectiveness to capture maneuvering targets by placing emphasis to current measurements. To this end, the proposal distribution in this paper for the 2-D position states is uniform over the FTA, and again the velocity states are zero.

The multiobject target density is then collapsed back into the PHD where

$$D(x_t|t) \approx \sum_{k=1}^{P_{ex}}\sum_{i=1}^{n^{(k)}} w_k^f \delta(x_t - x_{t,i}^{(k)}), \quad (17)$$
$$\approx \sum_{p^*=1}^{P^*} w_t^{(p^*)}\delta(x_t - x_t^{(p^*)}),$$

$\hat{N}_t \approx \sum_{k=1}^{P_{ex}} n^{(k)} w_k^f$, and $P^* = \sum_{k=1}^{P_{ex}} n^{(k)}$. Finally, the PHD is resampled back to P particles with uniform weights $w_t^{(p)} = \hat{N}_t/P$. The next stage of the filter continues with the prediction step in (15) at time t.

Initially at time 0, we generate NP 4-state samples to form P multiple target particles where $n^{(k)} = N$. The particles weights are computed by the likelihood update in (16), and then the particles are converted to P PHD particles to approximate $D(x_0|0)$. Given P PHD particles $\{w_{t-1}^{(p)}, x_{t-1}^{(p)}\}_{p=1}^{P}$ and $\hat{N}_{t-1} \approx \sum_{p=1}^{P} w_{t-1}^{(p)}$ at time $t-1$, the PHD filter proceeds by the following steps:

1. Obtain P predictive PHD particles: $x_{t-1}^{*(p)} = Fx_{t-1}^{(p)} + Q^{\frac{1}{2}}v^{(p)}, p = 1, 2, \ldots, P$.
2. Obtain P_{ex} Posisson samples with mean $\hat{N}_{t-1}$: $n^{(k)} \sim Poisson(\hat{N}_{t-1}), k = 1, 2, \ldots, P_{ex}$, where $P_{ex} = Pn_{ex} + J$.
3. Obtain Pn_{ex} predicative multitarget particles

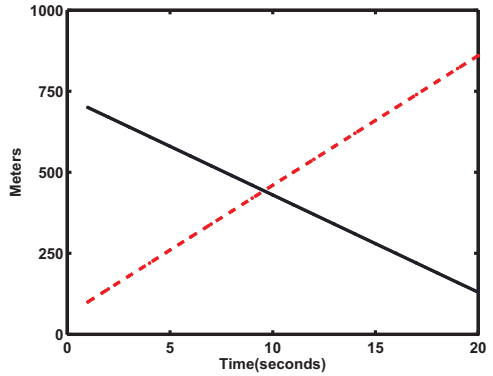
$$\{x_{t,1}^{(k)}, \ldots, x_{t,n^{(k)}}^{(k)}\}_{k=1}^{Pn_{ex}}$$

where the k-th multitarget particle consists of $n^{(k)}$ target states drawn from $\{x_{t-1}^{*(p)}\}_{p=1}^{P}$ with probability $w_{t-1}^{(p)}$.
4. Obtain J innovative multitarget particles

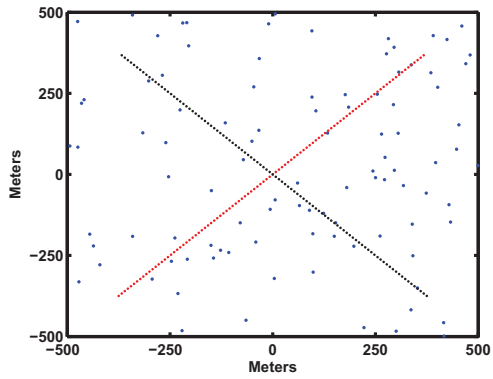
$$\{x_{t,1}^{(k)}, \ldots, x_{t,n^{(k)}}^{(k)}\}_{k=Pn_{ex}+1}^{Pn_{ex}+J}$$

where the k-th multitarget particle consists of $n^{(k)}$ target states drawn from a proposal density $p(\cdot|Y_t)$.
5. Obtain the weights $\{w_k^f\}_{k=1}^{P_{ex}}$ for the P_{ex} multitarget particles $\{x_{t,1}^{(k)}, \ldots, x_{t,n^{(k)}}^{(k)}\}_{k=1}^{P_{ex}}$ by (16).
6. Convert P_{ex} multitarget particles to P^* PHD particles at time t by (17).
7. Estimate the number of objects at time t: $\hat{N}_t \approx \sum_{k=1}^{P_{ex}} n^{(k)} w_k^f$.
8. Resample the P^* PHD particles $\{x_t^{(p^*)}\}_{p^*=1}^{P^*}$to obtain P particles $\{x_t^{(p)}\}_{p=1}^{P}$ where $x_t^{(p^*)}$ is selected with probability $w_t^{p^*}$.
9. Weigh each PHD particle $x_t^{(p)}$ uniformly as $w_t^{(p)} = \hat{N}_t/P$.
10. Let $t := t + 1$ and goto Step #1.

After the P particles for $D(x_t|t)$ is obtained, the k-means is used to group the particles into round($\hat{N}_t$) clusters. The centroids of these clusters are used as the estimated target states. The PHD avoids the track to measurement association problem that plagues most multi-target tracking methods. By avoiding the association problem, however, the output of the PHD filter is a list of points and not tracks. The PHD does not associate the points estimated at other points of time to the present set of points. For many applications, this does not matter. What matters is that the PHD incorporates the dynamic models of the targets, e.g., through the state transition and process noise covariance matrices, and the measurements to determine the likely target locations in a nearly optimal Bayesian sense.

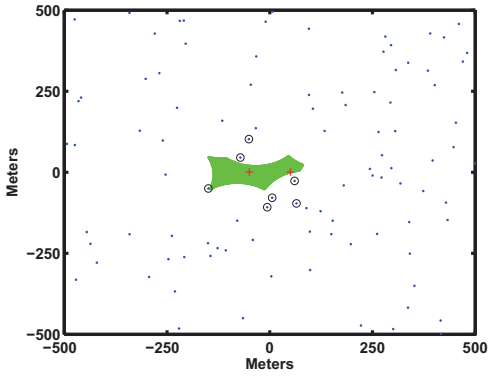


(a)

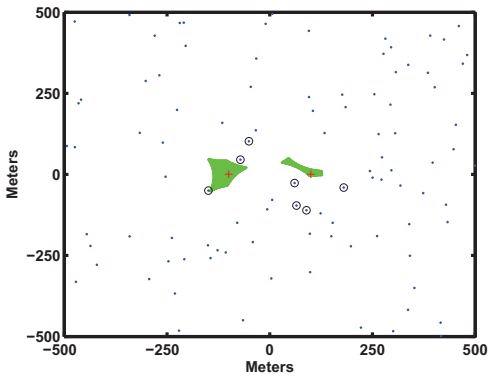


(b)

Figure 2. Geometry of the two scenarios: (a) Scenario 1: 1-D tracking showing the position of the two targets as a function of time, and (b) Scenario 2: 2-D tracking tracing out the positions of the two targets and plotting the positions of the 100 sensors.



(a)



(b)

Figure 3. Feasible target area for two targets when sensors follow the disc sensor model: (a) Targets are separated by 100m, and (b) targets are separated by 200m. The blue circles are the reporting detection sensors, shaded areas are the FTAs, and the plus symbols are the true target positions.

5. SIMULATIONS

The set of simulations considers a 1-D and 2-D tracking scenario. The geometries of the two scenarios are illustrated in Fig. 2. Fig. 2(a) plots the 1-D position of each target versus time for Scenario 1. The targets are traveling through a 1-D field of size 1km and they cross over each other halfway through the simulations at $t = 10$. Fig. 2(b) traces the 2-D positions of the two targets throughout the lifetime of the simulation. The targets are traveling through a 2-D field of size 1km $\times$ 1km, and they also cross over halfway through the simulations. In both scenarios, 100 sensors are randomly dispersed over the surveillance region.

To get a sense of of the FTA, Fig. 3 shows the FTA as a result of the detection reports. The blue dots represent sensor locations and the blue circles surround sensors that detect the targets. The green shaded region is the FTA, and the plus signs show the true target positions. In the figure, the disc sensing model is assumed where the sensing range is 117m. The sensors that do not detect the targets help to reduce the size of the FTA. In Figs. 3(a) and (b), the targets are separated by 100 and 200 meters respectively. It is clear that when the targets are closer together, the FTA tends to form a single connected region; whereas, for larger separation, the detections form separate clusters and the FTA forms multiple disconnected regions. While at least two targets could have lead to the FTA in Fig. 3(b), it unclear whether or not the simplest explanation for the FTA in Fig. 3(b) is one target in the center of the FTA or the ground truth of two targets separated by 100m. For the more realistic probabilistic sensing model, two targets would have increased the number of sensors detecting the targets. For the probabilistic model, a missed detection could remove one of the valid disconnected regions of a FTA. This can create problems with the ClusterTrack where the removal of such regions destroys the ability of ClusterTrack to lock onto targets.

For each scenario, we generate 100 random configurations of 100 proximity sensors, and for each configuration we run a single Monte Carlo realization of sensor reports. To understand what each method can and cannot do when the sensors behave via the two sensing model, we consider to cases for each scenario where either the sensors report detection based upon the disc or probabilistic sensing model. Note that for

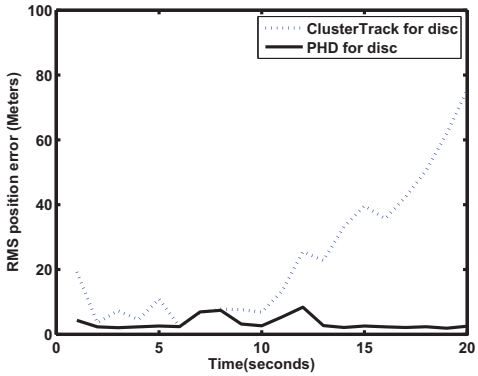
	K	L	H	m	$D_0(t)$
Scenario 1	500	2500	50	30	140t

Table 1. Parameters for ClusterTrack

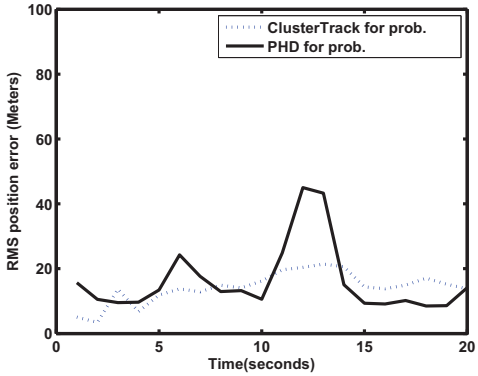
the disc model, the sensor reports are not random. In either case, the PHD uses the correct sensing model to update the weights in (16). Future work will investigate the effects of the robustness of the PHD to model mismatches. Overall for each scenario, the reported root mean squared (RMS) estimation errors for a given time is averaged over 100 simulations and the number of true targets. The RMS error for each target is reported by associating the closest k-means centroid for the PHD or the closest cluster head at time t for the ClusterTrack to the ground truth target location. This particular method penalizes the underestimation of the number of targets because the available estimated target locations will be far from the ground truth. However, when the number of targets is overestimated, there is a larger likelihood that one of the estimated target locations is close to the ground truth. It turns out that the number of cluster heads produced by the parameterization of ClusterTrack is much greater than two. Therefore, this particular method to estimate the the RMS position error tends to favor ClusterTrack. This section also reports the number of estimated targets for each method.

For Scenario 1, we run the PHD and ClusterTrack methods. In the disc model, $r_{eff} = 88$m, which is the distance that makes $P_d = 0.99$ for $P_{fa} = 0.001$ in the probabilistic model. We also use this range to compute the FTA. For the ClusterTrack, the parameters are shown in Table 5. The ClusterTrack uses the cost function (8) to rank particles and (10) to group similar particles in a cluster. Note that the threshold $D_0(t)$ used to determine the similarities of clusters is $140t$ which is chosen between $r_{eff}t$ and $2r_{eff}t$.

Fig. 4 shows the performance of the PHD and ClusterTrack as a function of time when the number of sensors is 100 for Scenario 1. Likewise, Fig. 5 shows the estimated number of targets as function of time. It appears that the position error for the PHD is lower for disc model than it is for the probabilistic model. However, this might be due to the bias caused by the fact that for the disc model the PHD tends to overestimate the number of targets. In the probabilistic model, the ClusterTrack is worse before the crossover than in the disc model but is able to recover after the crossove. Interestingly, for the disc model, ClusterTrack tends to loose track of a target after the crossover point at $t = 10$s. For the probabilistic model, the random false alarms and miss-detections seem to help out ClusterTrack at the crossover point. However, the crossover point poses some problems for the PHD for the probabilistic model. This may be caused by the fact that the PHD relies on the FTA to generate the innovations. While these innovations tend to help drive down the error in the disc case, it does not help very much for the probabilistic case where false alarms and miss-detections tend to put innovative target states in the



(a)



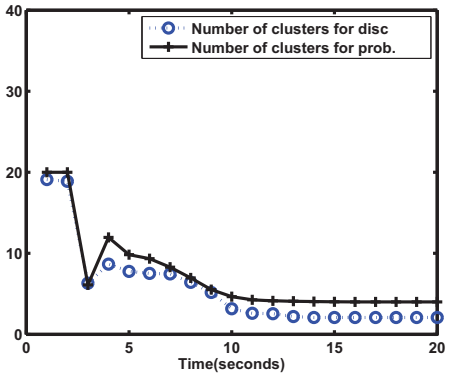
(b)

Figure 4. Scenario 1: RMS position errors of PHD and ClusterTrack as a function of time when the number of sensors is 100, and sensors follow the (a) disc or (b) probabilistic sensor model.

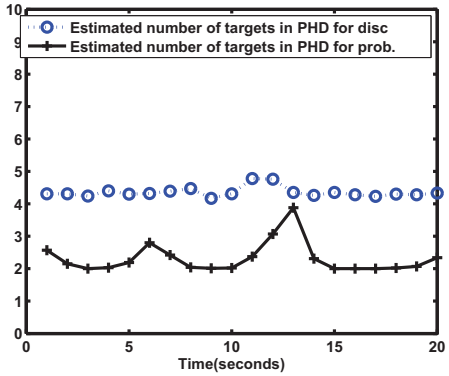
wrong place. Overall, the position accuracy of the PHD is much better than ClusterTrack for the disk model and comparable for the probabilistic model. On the other hand, the PHD can get a fairly good estimate for the number of targets for the probabilistic model, but there is no relation to the number of cluster heads and the number of targets for ClusterTrack.

The problems with ClusterTrack for the disc model after the crossover point are either caused by particles that are either dominated by one target or jump between two targets. These problems are demonstrated in Fig. 6 and Fig. 7. In Fig. 6, the cluster heads jump between two targets after the crossover point due to the FTA samples not covering all the targets at time 13 and time 15. In Fig. 7, the cluster heads are dominated by a single target after the crossover point due to how the particle are ranked. In this case, the FTA covers both targets but the particles associated to the second target never form cluster heads because their costs are to high. The ability for ClusterTrack to avoid single target dominance depends on how the clustering parameters allow low ranking particles associated to the other target to remain.

Next, we compare the PHD and ClusterTrack methods for



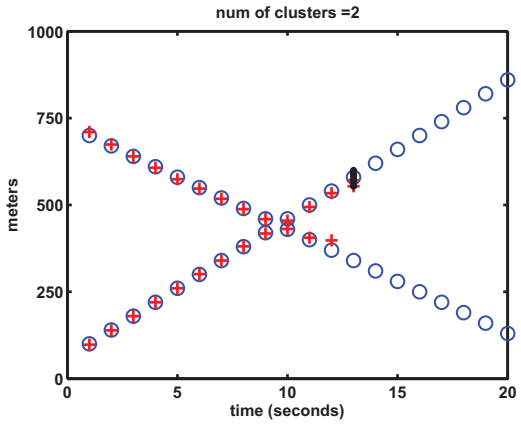
(a)



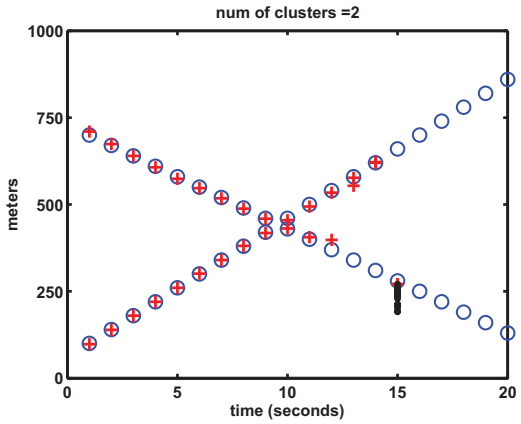
(b)

Figure 5. Scenario 1: Number of estimated targets as a function of time when the number of sensors is 100 and using (a) ClusterTrack or (b)PHD.

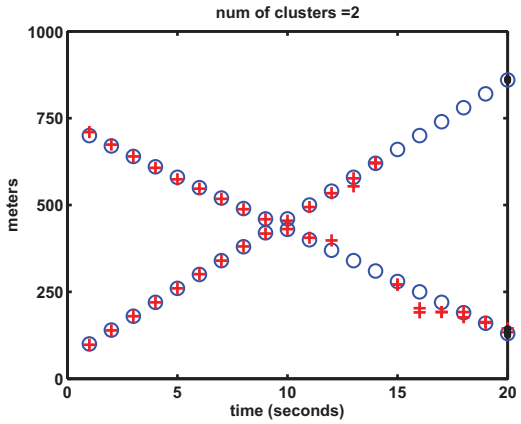
Scenario 2. In the disc model, $r_{eff} = 112$m, which is the distance that makes $P_d = 0.5$ for $P_{fa} = 0.001$ in the probabilistic model. We also use this range to compute the FTA. Instead of using (8), we use the cost function (9) to rank particles in the ClusterTrack method. This is attempt to avoid the dominating target problem. ClusterTrack tends to follow one target after the crossover point, which means that some particle experience an abrupt direction change over a short time. Although using either cost function is inevitable to suffer from "jumping between targets" problem due to lack of robustness on the parameters setup, we found that the cost function (9) experience less cases of target jumping than (8). Fig. 8 shows that the localization accuracy of the PHD is much better than the ClusterTrack in both sensor models. The ClusterTrack severely suffers from the "jumping between targets" problem in both sensor models. The plot for the PHD in Fig. 8(b) is similar to the results presented in [8], which uses a uniform proposal distribution over the entire surveillance region. In other words, using the FTA to produce innovative particles does not seem to help very much for the probabilistic sensing model. Finally, Fig. 9 plots the estimated number of targets using PHD and the number of clusters using ClusterTrack for both scenarios. As discussed, the number of clusters in the ClusterTrack does not provide reliable information for the tar-



(a)

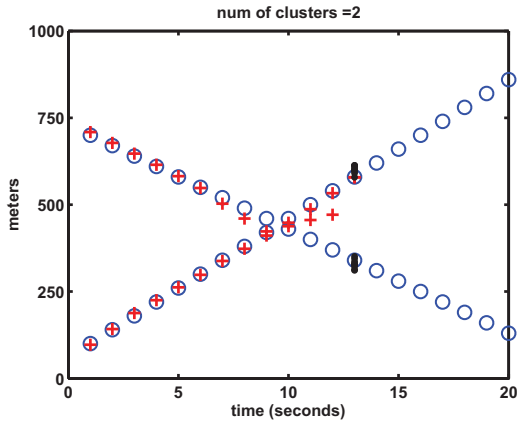


(b)

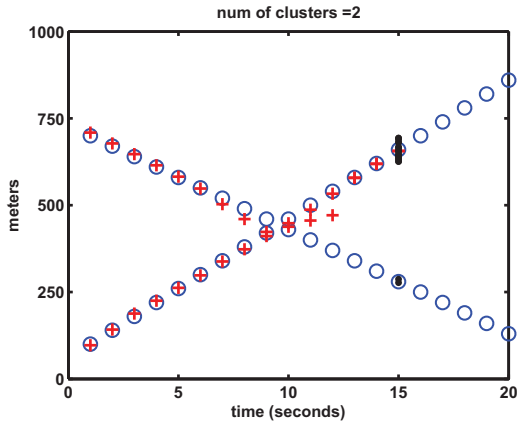


(c)

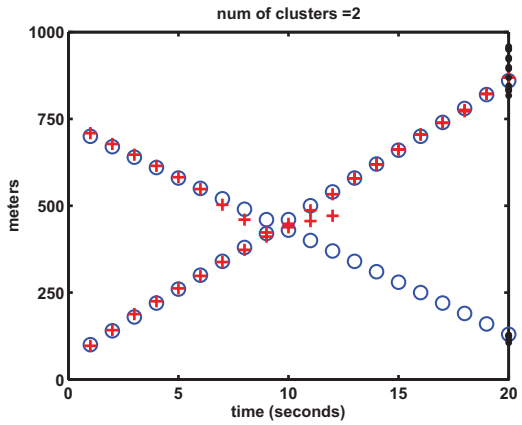
Figure 6. Scenario 1: ClusterTrack performance due to the missing FTA coverage of all targets when sensors follow the disc model (a) at time 13, (b) at time 15, and (c) at time 20, where the circles are true target positions, black dots are the random samples over the FTA, and the red pluses are the cluster head positions.



(a)

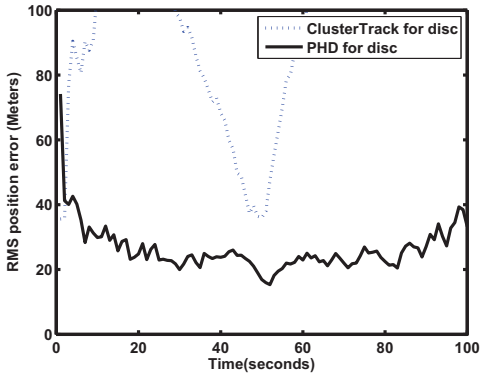


(b)

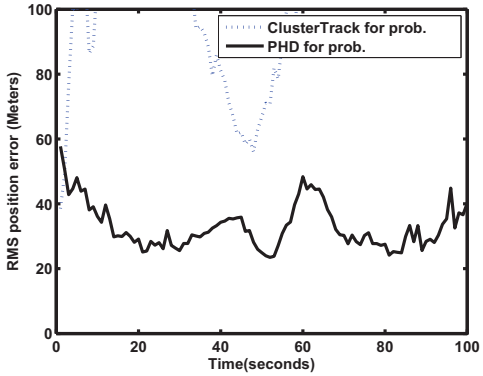


(c)

Figure 7. Scenario 1: ClusterTrack performance due to how particles are ranked when sensors follow the disc model (a) at time 13, (b) at time 15, and (c) at time 20, where the circles are true target positions, black dots are the random samples over the FTA, and the red pluses are the cluster head positions.



(a)



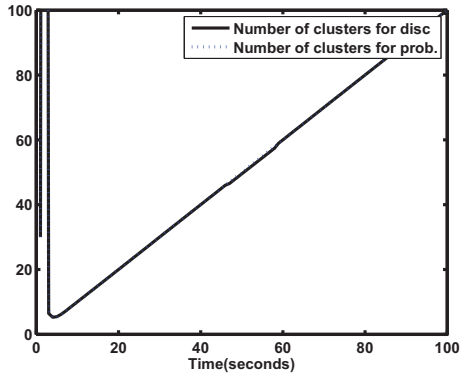
(b)

Figure 8. Scenario 2: RMS position errors of PHD and ClusterTrack as a function of time when the number of sensors is 100, and sensors follow (a) disc (b) probabilistic sensor model

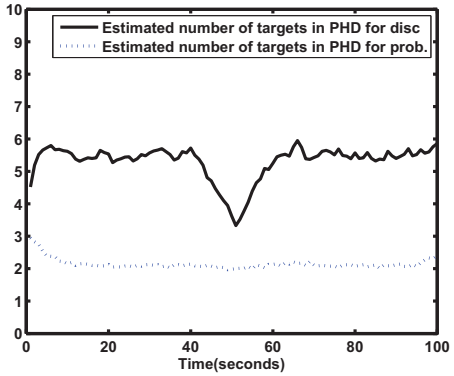
get count. It depends on how parameters are setup. Like in Scenario 1, the PHD overestimates the target count for the disc model due to the ambiguity, and is very accurate for the probabilistic model.

6. Conclusions

This work documents our investigation of multiple target tracking filters in proximity sensor networks. To this end, we implemented a particle-based PHD filter and compared it to ClusterTrack, a clustering-based particle filter. The simulations that compare the PHD to the ClusterTrack method demonstrate the advantages of PHD method. For the 2-D tracking case, the localization accuracy of the PHD is far better than that of the ClusterTrack. Unlike ClusterTrack, the PHD is able to estimate the number of targets. The advantages of the PHD may be due to the fact that it is based principled Bayesian approach; whereas the ClusterTrack is more ad-hoc in its design. Future work will consider improvements to the PHD filter implementation to obtain better innovative particles for the probabilistic model. Furthermore, we plan to investigate the robustness of the PHD to model mismatches and to parameter selection. Finally, we hope to be able to characterize the performance of the PHD as a function of sen-



(a)



(b)

Figure 9. Scenario 2: Number of estimated targets as a function of time when the number of sensors is 100, using (a) ClusterTrack (b)PHD

sor density, sensor sensitivity and target separation.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. of IEEE ICASSP 2001*, vol. 4, May 2001, pp. 2033–2036.

[2] A. Artes-Rodriguez, M. Lazaro, and L. Tong, "Target location estimation in sensor networks using range information," in *IEEE Sensor Array and Multichannel Signal Processing workshop*, 2004.

[3] P. Djuric, M. Vemula, and M. Bugallo, "Target tracking by particle filtering in binary sensor networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2229–2238, 2008.

[4] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target tracking with binary proximity sensors," *ACM Trans. on Sensor Networks*, vol. 5, no. 4, Nov. 2009.

[5] Z. Wang, E. Bulut, and B. K. Szymanski, "Distributed energy-efficient target tracking with binary sensor networks," *ACM Trans. on Sensor Networks*, vol. 6, no. 4, Jul. 2010.

[6] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking multiple targets using binary proximity sensors," in *Proceedings of IPSN*, 2007.

[7] Q. Le and L. M. Kaplan, "Target localization using proximity binary sensors," in *Proc. of the IEEE Aerospace Conference*, Big Sky, MT, Mar. 2010.

[8] ——, "Target localization using proximity binary sensors," in *Proc. of the 27th Army Science Conference*, Orlando, FL, Nov. 2010.

[9] R. Mahler, "Multi-target bayes filtering via first-order multi-target moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[10] D. E. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the phd filter," in *Proc. of the IEE Radar, Sonar and Navigation*, vol. 152, 2005, pp. 327–334.

[11] D. E. Clark, I. Ruiz, Y. Petillot, and J. Bell, "Particle phd filter multiple target tracking in sonar image," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 409 –416, january 2007.

[12] Y.-D. Wang, J.-K. Wu, A. Kassim, and W. Huang, "Data-driven probability hypothesis density filter for visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1085 – 1095, 2008.

[13] M. Tobias and A. Lanterman, "Probability hypothesis density-based multitarget tracking with bistatic range and doppler observations," in *Proc. of the IEE Radar, Sonar and Navigation*, vol. 152, 2005, pp. 195 – 205.

[14] B. T. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multi-target filtering with random finite sets," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[15] B. T. Vo, B. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. on Signal Processing*, vol. 55, no. 7, pp. 3553–3567, 2007.