Probability Hypothesis Density Filter for Multitarget Multisensor Tracking

O. Erdinc, P. Willett, Y. Bar-Shalom
ECE Department
University of Connecticut
ozgur, willett @ engr.uconn.edu
ybs@ee.uconn.edu

Abstract – Multiple target tracking techniques require data association that operates in conjunction with filtering. When multiple targets are closely spaced, the conventional approach (MHT/assignment) may not give satisfactory results, mainly due to the difficulty in deciding the number of targets. Recently, the first moment of the “multi-target posterior density”, called the probability hypothesis density (PHD), has been proposed to address the multi-target tracking problem. Particle filtering techniques have been applied to implement the PHD based tracking.

In this paper, we explain our interpretation of the PHD, then investigate its performance on the problem of tracking unresolved targets from multiple sensors. In the set-up, there are two different radars which monitor the targets, and the PHD is fed sequentially by these scans.

In the scenario, we investigate 3 different levels of complexity in terms of measurement extraction methodologies of sensors when there are unresolved targets 1. Sensor model reports a measurement with variance \( \sigma_{\text{mono}}^2 \). (Sensor is not capable of sensing any abnormality in radar return). 2. Sensor model gives a single measurement with a larger variance \( \sigma_{\text{azi}}^2 \geq \sigma_{\text{mono}}^2 \). 3. Sensor model uses a multi-target measurement extractor. Unresolved targets create separate measurements with variance \( \sigma_{\text{mono}}^2 \).

Simulation results for two-dimensional scenario are given to show the performance of the approach. Based on our simulation results, we also discuss difficulties the PHD algorithm seems to encounter, especially as is reflected in the target “death” event.

Keywords: Unresolved targets, probability hypothesis density filter, PHD, multi-target tracking, particle filter.

1 Introduction

Recently, probability hypothesis density filter (PHD) has been proposed; there are many citations, and [8, 9] seem representative. The PHD filter propagates the PHD or the first moment of the multi-target posterior density where the integral of the PHD over a region in the state space is the expected number of targets within this region, and the peaks in the PHD can be regarded as the estimated locations of the targets at a given time step.

However, there are some issues related to the implementation of the PHD filter. Since the PHD propagation equations involve multiple integrals, there are no computationally tractable exact closed form expressions. In addition, multi-peak extraction becomes a critical issue to the successful implementation of a PHD filter. Consequently, different techniques for sequential Monte Carlo implementation of the PHD filter for multi-target tracking have been proposed [7, 12, 13, 14].

The Integrated Probabilistic Data Association (IPDA) [10] addresses the issue of target existence as well as the estimation of target states. The fundamental linkages between Random Finite Sets(RFS) formalism and IPDA algorithm have been established in [4]. In that paper, it is shown that IPDA can be derived from the RFS formalism (although not from the PHD) in a straightforward manner.

In this paper, we investigate the performance of the PHD filter in multi-sensor case where there are unresolved targets. We also look at different sensor models, where the sensors’ behaviors differ in the case that there are more than one (unresolved) target in a resolution cell.

2 Probability Hypothesis Density (PHD)

2.1 Probability Hypothesis Density

Our presentation of PHD will be slightly different than the previous publications [9, 14]. Besides the fact that the PHD is first-order factorial moment of a finite point process [5, 9], it also can be thought as the probability that there is a target in a given infinitesimal region of \( S \).

We first start with the definitions,

\[
d : \text{dimension of the state vector } x
\]
\[ S \subset \mathbb{R}^d \quad : \text{surveillance region} \]
\[ x \in \mathbb{R}^d \quad : \text{a point in } \mathbb{R}^d \]
\[ (x + dx) \subset S \quad : \text{an infinitesimal region around the point } x \text{ in } S \]
\[ \pi_n(x_1, \ldots, x_n) \quad : \text{the joint probability density of targets’ states, conditioned on there are } n \]
\[ p_n \quad : \text{probability that there are } n \text{ targets} \]

Hence, \( \pi_n(.) \) represents a proper (integrates to unity) probability density function where it is assumed that targets are identified, i.e., “labelled”. From the perspective of multi-target tracking algorithms, \( \pi_n(.) \) is naturally permutation symmetric, i.e., supposing \( n = 3 \),

\[ \pi_3(x_1, x_2, x_3) = \pi_3(x_2, x_1, x_3) = \ldots = \pi_3(x_3, x_2, x_1) \quad (1) \]

If we assume there are \( n \) targets in the scene, we can express the probability density at \( x, - \) or the probability that region \((x, x + dx)\) has a target – as the summation of marginal densities,

\[
\text{PHD}(x|n) = \int \pi_n(x, y_1, \ldots, y_{n-1}) dy + \int \pi_n(y_1, x, y_2, \ldots, y_{n-1}) dy + \ldots + \int \pi_n(y_1, \ldots, y_{n-1}, x) dy \quad (2)
\]

\[
= n \int \pi_n(x, y_1, \ldots, y_{n-1}) dy \quad (3)
\]

where \( dy = dy_1 dy_2 \ldots dy_{n-1} \). Then, the PHD(\( x \)) is,

\[
\text{PHD}(x) = \sum_{n=0}^{\infty} (n+1)p_{n+1} \int \pi_{n+1}(x, y_1, \ldots, y_n) dy \quad (4)
\]

\[
= \sum_{n=0}^{\infty} \frac{1}{n!} \int f_{k|k}(\{x, y_1, \ldots, y_n\}|Z^k) dy \quad (5)
\]

where \( dy = dy_1 dy_2 \ldots dy_n \) and \( f_{k|k}(\cdot) \) is multi-target posterior likelihood as defined in [8, 9]. Equivalently, the PHD can be expressed as set integrals

\[
\text{PHD}(x) = \int f_{k|k}(\{x\} \cup Y) \delta Y \quad (6)
\]

\[
= \sum_{n=0}^{\infty} \frac{1}{n!} \int f_{k|k}(\{x\} \cup \{y_1, \ldots, y_n\}) dy \quad (7)
\]

The PHD filter propagates and updates this probability density surface by using the current measurement set \( Z^k \). Hence, integration of the PHD in a given region gives expected number of targets therein. The filter recursion is given in [9]. The predicted PHD is (by introducing time indices \( k \) and \( k-1 \))

\[
D_{k|k-1}(y|Z^{k-1}) = b_{k|k-1}(y) + \int D_{k|k-1}(y|x) D_{k-1|k-1}(x|Z^{k-1}) dx \quad (8)
\]

where

\[
D_{k|k-1}(y|x) = (1 - d_{k|k-1}(x)f_{k|k-1}(y|x) + b_{k|k-1}(y|x) \quad (9)
\]

and

\[
b_{k|k-1}(y) \quad : \text{PHD of the spontaneous target birth} \]
\[
d_{k|k-1}(x) \quad : \text{probability of target death} \]
\[
f_{k|k-1}(y|x) \quad : \text{transition probability density} \]
\[
b_{k|k-1}(y|x) \quad : \text{PHD of targets spawned by existing targets} \]

Given the new scan of data \( Z_k = \{z_1, \ldots, z_m\} \), the updated PHD

\[
D_{k|k}(x|Z^k) = \sum_{z \in Z_k} \left[ \frac{P_d D_k(z)}{\lambda_k c_k(z)} + \frac{P_d D_k(z)}{\lambda_k c_k(z)} D_k(x|z) \right] + (1 - P_d) D_{k|k-1}(x|Z^{k-1}) \quad (10)
\]

where

\[
D_k(z) = \int f_k(z|x) D_{k|k-1}(x|Z^{k-1}) dx \quad (11)
\]

\[
D_k(x|z) = \frac{f_k(z|x)}{D_k(z)} D_{k|k-1}(x|Z^{k-1}) \quad (12)
\]

\[
P_d \quad : \text{probability of detection} \]
\[
\lambda_k \quad : \text{average number of false alarms per scan, assumed Poisson distributed} \]
\[
c_k(z) \quad : \text{spatial density of false alarms} \]
\[
f_k(z|x) \quad : \text{sensor likelihood function} \]

The estimated number of targets is given by integration of the PHD over all surveillance region

\[
T_{k|k} = \int_S D_{k|k}(x|Z^k) dx \quad (13)
\]

Estimation of the targets’ states can be done by searching for the peaks of the PHD surface. \([T_{k|k}]\) largest peaks of \( D_{k|k}(x|Z^k) \) correspond to those targets’ locations (states), where \([T_{k|k}]\) denotes the nearest integer to \( T_{k|k} \).

3 Implementation of the PHD

The PHD propagation (8)-(10) have no computationally tractable closed form expressions even for the simple case where individual targets follow a linear Gaussian dynamic.
model. Particle filtering techniques permit recursive propagation of the full posterior [6] and have been used for near-optimal Bayesian filtering.

In recent papers of [12, 13, 14] the particle filter implementation of the PHD filter is given in great detail. Hence, we will mostly stress the implementation differences in our approach, then discuss the results.

3.1 Kinematic Model

In our simulations, a two-dimensional discrete time kinematic model is considered and linear Gaussian dynamics in the x-y coordinate frame are assumed. The states of the targets consist of position and velocity, while only position measurements are obtained by sensors. The true target state is

\[
x(k) = [x(k) \ y(k) \ \dot{x}(k) \ \dot{y}(k)]' \quad (14)
\]

where \((x(k), y(k))\) and \((\dot{x}(k), \dot{y}(k))\) denote target position and velocity at time step \(k\). The evolution of the target state is

\[
x(k+1) = Fx(k) + v(k) \quad (15)
\]

where \(\{v(k), k = 0, 1, \ldots\}\) is the sequence of white Gaussian process noises, with covariance \(Q(k)\), and

\[
F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)
\]

in which \(T\) denotes the sampling interval.

For each radar scan, each measurement consists of range and bearing (azimuth)

\[
z = \begin{bmatrix} r_m \\ \theta_m \end{bmatrix} = \begin{bmatrix} r + w_r \\ \theta + w_\theta \end{bmatrix} \quad (17)
\]

where \(r\) and \(\theta\) are the true range and bearing, and \(w_r\) and \(w_\theta\) are independent zero-mean measurement noises with

\[
E[w_r^2] = \sigma_r^2 \quad E[w_\theta^2] = \sigma_\theta^2 \quad (18)
\]

The standard conversion \(^2\) to Cartesian is

\[
x_m = r_m \cos \theta_m \quad y_m = r_m \sin \theta_m \quad (19)
\]

with (approximate) covariance terms based on linearization

\[
\text{var}(x_m) = r_m^2 \sigma_\theta^2 \sin^2 \theta_m + \sigma_r^2 \cos^2 \theta_m \quad (20)
\]

\[
\text{var}(y_m) = r_m^2 \sigma_\theta^2 \cos^2 \theta_m + \sigma_r^2 \sin^2 \theta_m \quad (21)
\]

\[
\text{cov}(x_m, y_m) = (\sigma_r^2 - r_m^2 \sigma_\theta^2) \sin \theta_m \cos \theta_m \quad (22)
\]

\(^1\)Note that the original particle PHD filter is more general for handling non-linear & non-Gaussian situations. Here we concentrate on the linear Gaussian case.

\(^2\)This conversion is biased. However the bias is negligible in our case that the sensor beam-width is small [1].

and the covariance matrix for the measurement \(z\) is

\[
R(z) = \begin{bmatrix} \text{var}(x_m) & \text{cov}(x_m, y_m) \\ \text{cov}(x_m, y_m) & \text{var}(y_m) \end{bmatrix} \quad (23)
\]

We assume that the radars are using the (real part of the) monopulse ratio to extract the angle of arrival (AOA). As discussed in [3], the monopulse ratio provides angular estimates that are biased by a multiplicative amount whose magnitude decreases as the signal-to-noise ratio (SNR) increases. Hence, with each AOA measurement \(\theta_m\), the corresponding (observed) SNR is reported to the PHD filter and this varying \(\sigma_\theta^2\) is used to calculate \(R(z)\).

3.2 Particle Implementation of the PHD

The procedure of implementing the algorithm is as follows. At \(k = 1\), we initialize \(N_z\) particles around each measurement. Position components of these particles are drawn from a Gaussian distribution, \(N(z, R(z))\), and the velocity components are drawn from a uniform distribution \(U[-\nu_{\text{max}}, \nu_{\text{max}}]\). Each particle consists of two elements: a sample from the state space, \(\xi^{(i)}\), and its corresponding weight, \(w^{(i)}\), for \(i = 1, \ldots, N\) where \(N\) is the total number of particles proposed. All these particles are given equal weights, which of course sum to the initial guess for the number of targets.

For \(k \geq 2\), we proceed as follows:

**Step 1: Prediction step**

Assuming we have \(N\) particles at time \(k - 1\), the particles propagate according to the motion model with additive white Gaussian noise, namely,

\[
\xi_{k|k-1}^{(i)} = F \xi_{k-1}^{(i)} + v(k) \quad (24)
\]

and the corresponding predicted weights

\[
w_{k|k-1}^{(i)} = w_{k-1}^{(i)} P_s \quad (25)
\]

for \(i = 1, \ldots, N\), where \(P_s\) is the probability of target survival.

**Step 2: Additional Particle Proposal**

For each \(z \in Z_k\), we sample \(N_z\) particles around the measurement. Position components of these particles are drawn from a Gaussian distribution, \(N(z, R(z))\), and the velocity components are drawn from uniform distribution \(U[-\nu_{\text{max}}, \nu_{\text{max}}]\). Equal weights are assigned for these additional particles, while ensuring that they sum to the PHD of target birth for the whole surveillance region, \(b_{k|k-1}(S)\),

\[
w_{k|k-1}^{(i)} = \frac{b_{k|k-1}(S)}{N_{\text{new}}} \quad (26)
\]

where \(N_{\text{new}} = N_z |Z(k)|\) is the total number of particles proposed for the “investigation” of newborn targets.
Step 3: Update step
For each $z \in Z_k$, we compute

$$C(z) = \sum_{j=1}^{N+N_{\text{new}}} P_d f(z|\xi^{(i)}) w_k^{(j)}$$

(27)

for $i = 1, \ldots, N + N_{\text{new}}$. Then, the weights are updated by

$$w_k^{(i)} = \left[1 - P_d + \frac{1}{\sum_{z \in Z} \lambda_i c_k(z) + C(z)} \right] w_k^{(i)}$$

(28)

Step 4: Peak Extraction
To estimate target states, the updated PHD is approximated by a Gaussian Mixture by using the expectation-maximization (EM) algorithm [2] — a Gaussian mixture is a common choice for peak extraction among recent implementations of the PHD [12, 13]. The EM algorithm fits $M$ Gaussian pdfs (probability density functions) to the PHD surface and gives corresponding weights to each of these to indicate the quality of fitting. Then, the mean and the covariances of the $\{T_k\}_{j=1}^{N_{\text{new}}} \{\xi^{(i)}\}$ heaviest Gaussian pdfs in the mixture are denoted as peaks at time $k$. The EM algorithm steps, modified to incorporate the weights $w_k^{(i)}$ of the PHD, are given next.

$$\delta_{im} = \frac{1}{\sqrt{2\pi\Omega_m}} e^{-\frac{1}{2} (\xi^{(i)} - \mu_m)^T \Omega_m^{-1} (\xi^{(i)} - \mu_m)}$$

(29)

$$\delta_{im} = \frac{w_k^{(i)} \delta_{im}}{\sum_{l=1}^{M} \delta_{il}}$$

(30)

$$\mu_m = \frac{\sum_{i=1}^{N+N_{\text{new}}} \xi^{(i)} \delta_{im}}{\sum_{i=1}^{N+N_{\text{new}}} \delta_{im}}$$

(31)

$$\Omega_m = \frac{\sum_{i=1}^{N+N_{\text{new}}} \delta_{im} (\xi^{(i)} - \mu_m) (\xi^{(i)} - \mu_m)^T}{\sum_{i=1}^{N+N_{\text{new}}} \delta_{im}}$$

(32)

In order to have a common covariance for all Gaussian pdfs (the “homoscedastic” version of the EM algorithm), we use

$$\Omega = \frac{\sum_{i=1}^{N+N_{\text{new}}} \sum_{m=1}^{M} \delta_{im} (\xi^{(i)} - \mu_m) (\xi^{(i)} - \mu_m)^T}{\sum_{m=1}^{M} \sum_{i=1}^{N} \delta_{im}}$$

where $\delta_{im}$ denotes the weight, $\mu_m$ denotes the mean value and $\Omega_m$ denotes the covariance of the $m^{th}$ Gaussian of the mixture, for $m = 1, \ldots, M$. The EM algorithm starts with initial values of $\kappa_m, \mu_m$ and $\Omega_m$, and repeats steps (29) → (32) until it converges.

The algorithm fits $M = |Z(k)| + |T_{k-1}|$ Gaussian pdfs at each scan. The initial mean values are chosen as the measurement locations $Z(k)$ and at the peaks at previous scan ($k - 1$). We use the homoscedastic version until the algorithm converges, since homoscedastic Gaussian mixture is generally quite stable. However, we then make 3 more runs to get diverse covariances.

That is, we initialize with a homoscedastic Gaussian mixture. Then we invoke a heteroscedastic Gaussian mixture algorithm, such that the covariances can be different: some targets will be precisely located by the measurements and algorithm, such that their peaks ought to be concentrated; and others with the opposite behavior.

Step 5: Resampling
Particles $\{w_k^{(i)} , \xi^{(i)}\}_{i=1}^{N+N_{\text{new}}}$ are resampled according to a Monte Carlo technique, so that each particle is resampled proportionally to its weight, while preserving the total weight as $T_{k|k}$, which is computed by.

$$T_{k|k} = \sum_{j=1}^{N+N_{\text{new}}} w_k^{(j)}$$

(33)

After resampling each particle is given equal weight.

4 Simulation Results
4.1 Scenario
In our scenario, two targets move in a 200km-by-200km surveillance region. One of the targets splits (spawns) into two at $t = 200$[seconds]$^3$, where the beginning of the scenario is taken as $t = 0$. Target 1 heads to south-east approximately at speed Mach 1 and Target 2 heads to north-east with the same speed ($\text{vel}_1 = 350\text{m/s}$, $\text{vel}_2 = 50\text{m/s}$). The splitting target has an initial acceleration of $1\text{m/s}^2$.

Two sensors monitor these targets. Sensor 1 (S1) is located at (0 km;0 km) and the other (S2) is at (200 km;0 km). The splitting target accelerates towards the north. Sensor S2 is more advantageous than S1, since S2 sees targets towards the beam, whereas S1 sees them across-the-beam. (Figure 1) Due to its geometrical disadvantage Sensor 1 sees the targets resolved (latest) at $t = 248$, while S2 sees them at $t = 226$.

In the simulations, the number of (additionally) proposed particles for each measurement is $N_z = 1000$ for the first scan $k = 1$ and $N_z = 100$, for $k \geq 2$. $^4 v_{\text{max}} = 500$ [m/s]. The average number of clutter measurements $\lambda_k = 2 / \ell$

$^3$The time index $t$ [seconds] is preferred in this section, whereas $k$ is used for scans throughout the paper.

$^4$With realistic scenario parameters, e.g., target velocities, it becomes obvious that the particles proposed in Step 2 (from $U(-v_{\text{max}}, v_{\text{max}})$) turn out to be inadequate to represent a new-born-target’s velocity.

In our scenario, fortunately, this difficulty is overcome by using an excessive number of particles ($N_z = 1000$) only at first scan. Since the new-born target in fact splits from an existing one, the particle representation of the PHD at split time is good enough to inform this target. However, this motivates that an efficient particle proposal technique should be used. Such techniques has recently been proposed in [7, 11].
surveillance region]. $P_d$ is chosen as 1; although this is unrealistic, it facilitates our analysis (PHD’s $T_{k|k}$ estimates on unresolved targets). Probability of survival, $P_s = (1 - d)$, is 0.99. PHD of birth is $b_{k|k-1}(S) = 10^{-7}$ [1 / surveillance region]. The process noise variance for the motion model is chosen $E[σ^2_v] = 25$ [m/s$^2$]. We have $σ_θ = 0.5$ [degrees] and $σ_r = 60$ [meters], and nominal SNR of 20dB.

We assume 3 different sensor models, and perform separate Monte Carlo runs for each of them. All three of the models assume that radars are monopulse, however they differ when there are unresolved targets:

Model 1 – produces a single measurement at each scan where there are two targets in a resolution cell. However, it reports a larger azimuth variance ($σ_{azi} > σ_{mono}$) to indicate an abnormality in the radar return.

Model 2 – yields a single measurement with the usual monopulse azimuth variance ($σ_{mono}$).

Model 3 – represents the lower bound. The sensors report two different measurements with monopulse variances ($σ_{mono}$) as if targets are perfectly resolved.

Briefly;

M1 : $\{Z_1 |σ_{azi} > σ_{mono}\}$
M2 : $\{Z_1 |σ_{mono}\}$
M3 : $\{Z_1, Z_2 |σ_{mono}\}$

For models 1 and 2, the closely-spaced targets become resolved based on resolution probabilities, $P_{rec}^R$ and $P_{rec}^X$:

$$P_{rec}^R = \min\left\{ \frac{ΔR}{R}, 1 \right\}$$ (34)

$$P_{rec}^X = \min\left\{ \frac{ΔXR}{XR}, 1 \right\}$$ (35)

where

$ΔR$ : Distance between the targets in range
$ΔXR$ : Distance between the targets in cross-range
$R$ : Resolution cell length in range
$XR$ : Resolution cell length in cross-range

4.2 Results - Two Sensors

Both Sensor 1 and Sensor 2 have a 2-second sampling rate ($T_{samp} = 2$). However, we assume that they are synchronous and exactly ut-of-phase, and hence that the PHD filter receives a scan from alternating sensors every second. The PHD filter runs between $t = 190$ and $t = 270$, i.e., between scans 95 and 135. We are mainly interested in the estimated number of targets, $T_{k|k}$. In the figures 2 and 3, $T_{k|k}$ is plotted after PHD filter updates its estimation based on the observations received from Sensor 2, and observations received from Sensor 1, consequently.

4.3 Results - Only Sensor 1

In this case, Sensor 2 is not functioning, and the sampling rate for Sensor 1 is assumed $T_{samp} = 1$sec. Targets split at $t = 200$ and resolution probabilities become 1 at $t = 248$. 
In figure 4, the estimated number of targets for different sensor models are plotted. In addition to these results the average number of non-clutter measurements is given in a (dashed) plot. This is the ground-truth for the number of targets in the scene.

4.4 Results - Only Sensor 2

Similar to the previous, it is assumed that Sensor 1 is not functioning, and instead Sensor 2 monitors targets with a sampling rate of 1 second. The targets become resolved with probability one at \( t = 226 \). Figure 5 shows that the splitting target is detected earlier than the case where only Sensor 1 is used. Intuitively, this is expected since S2 has an advantage due to geometry.

It is not intuitively clear why “Only S2” detects the third target better (earlier) than the case in which two sensors used at the same time (compare Figures 2 and 5). The first comment on this is that the sensor we use in “Only S2” case is a more informative sensor, since its sampling rate is twice as high as S2 used in the “two sensors” case. Hence, we take a look at the case where only S2 is used but its sampling rate is two seconds. As seen in the Figure 6, the results are not greatly different. The increased sampling rate resulted in a second delay in the detection of the third target. Then, why is it that we cannot gain from using two sensors? We will come back to this question at the end of the next section.

5 A simpler scenario and the PHD

Recently, Challa et al. showed that the IPDA algorithm [10] can be formulated within the FISST framework under some assumptions of linearity [4]. Motivated by this, we will consider a simple case where there is only one target in the surveillance region and no false alarms are observed. The target moves in a straight line in the 2-dimensional plane. As in the previous simulations, the linear-Gaussian assumption holds.

Recall that if there is no measurement at the current scan, the PHD update equation (10) becomes

\[
D_{k|k}(x|Z^k) = (1 - P_d)D_{k|k-1}(x|Z^{k-1}) \quad (36)
\]

Hence, we will concentrate on the case in which there is a target in the scene for a while and then a missed detection occurs.

5.1 Simulation Results

In the simulation, the target is detected at scans 1 through 9, 15 and 20 through 30. There are no measurements observed during scans 10-14, and 16-19. Parameters are: \( P_t = 0.9, P_s = 0.8 \), Birth PHD = 0.15.
As seen in Table 1, the estimated number of targets, $T_{k|k}$, is confidently close to unity (which is the truth) until scan 10. With the first missed detection at scan 11, it drops down to 0.08, i.e. $(1 - P_d) P_{k|k}$. As long as there is no detection, it decreases by the factor 0.08 at each scan. At scan 15, with a detection, $T_{k|k}$ jumps back to 1 and at scan 16, it drops back to 0.08 and so on. This indicates that the PHD has little “memory”, at least as far as is reflected in target death. Hence it is expected to see the same kind of behavior at scan 10 and scan 16. Intuitively, one would expect to see a gentler decrease in the expected number of targets when a miss detection occurs, as happens in the IPDA.

### 5.2 What is wrong?

Considering a Markov Chain model for the target existing probability, $P\{X_{k|k}\}$, it can be shown that [4, 10],

$$
P\{X_{k|k}\} = \frac{(1 - P_d)q}{1 - P_dq} \quad (37)
$$

where $P\{X_{k|k-1}\} = q$, and probability of detection is $P_d$. Note that the prior probability is the prediction PHD in our scenario. With the parameters used in our simulation, assuming $T_{k-1|k-1} = 1$,

$$
P\{X_{k|k}\} = \frac{(1 - 0.9)0.8}{1 - 0.9 \times 0.8} = 0.28
$$

So, it would be reasonable if we did observe $T_{k|k} = 0.28$ instead of 0.08, when there is a missed detection.

Now, by taking the derivative with respect to $q$ in (37), we get

$$
\frac{d}{dq} P\{X_{k|k}\} = \frac{1 - P_d}{(1 - P_dq)^2} \quad (38)
$$

Clearly, this is equal to $(1 - P_d)$ when the prior target existence probability, $q$, is small – see Figure 7.

Figure 7: The slope of the curve is $1 - P_d$ when $q$ is small.

Consider an infinitesimal region in the state space. The PHD value, the expected number of targets in this region, is very small. Consequently, the update equation becomes (36), which is the PHD update when there are no measurements. This implies that the algorithm, in fact, does not “believe” a priori that there is a target in the scene, even though the PHD sums up to 1 over the whole surveillance region. This would seem to be a possible source of concern to PHD users.

### 5.3 Target-death problem in two sensor case

Here we consider the scenario used in Section 5.1. As described in this section, each sensor scans every 2 seconds, however, there is one second shift between them, so that the data is fed to the PHD filter every second. In Figure 8, between $t = 215$ and $t = 246$ the Sensor 1 sees the closely-spaced targets unresolved while Sensor 2 sees them resolved (to be precise, this happens with high probability, since resolutions probabilities for Sensor 2 is slightly higher while the PHD sums up to 1 over the whole surveillance region). This would seem to be a possible source of concern to PHD users.

| Scan no | Detection | $T_{k|k}$ |
|---------|-----------|-----------|
| 1       | Yes        | -         |
| 2       | Yes        | 1.095     |
| 3       | Yes        | 1.1026    |
| 4       | Yes        | 1.1032    |
| 5       | Yes        | 1.1033    |
| 6       | Yes        | 1.1033    |
| 7       | Yes        | 1.1033    |
| 8       | Yes        | 1.1033    |
| 9       | Yes        | 1.1033    |
| 10      | No         | 0.088261  |
| 11      | No         | 0.0070609 |
| 12      | No         | 0.00056487|
| 13      | No         | 4.519e-005|
| 14      | No         | 3.6152e-006|
| 15      | Yes        | 1.015     |
| 16      | No         | 0.0812    |
| 17      | No         | 0.006496  |
| 18      | No         | 0.00051968|
| 19      | No         | 4.1574e-005|
| 20      | No         | 3.326e-006|
| 21      | Yes        | 1.015     |

Table 1: Single Target - Estimated number of targets.
measurement from Sensor 1. In other words, S1’s belief that there are 2 targets implies that there is a missed detection for the third target, and $(1 - P_d)$ update clears the existent third target from the PHD surface.

Revisiting the question we asked in the Results section, (Figure 6): Why would the two-sensor case be less informative than the “Only S2” (or the single sensor, whichever has an advantage due to its geometrical orientation. It could be the S1 in another scenario.) case? Sensor S1’s belief that there are two targets provides counter-evidence to the observations of S2. Since S1 is taken more informative than it should be (it is a miss detection), the outcome of using two sensors is almost as bad as using “Only S1”. On the other hand, S2’s estimation on number of targets is consistent, and each observation supports the previous ones and builds up the evidence as fast as possible, i.e., as fast as its look to the scenario allows.

6 Conclusions

We have discussed the performance of the PHD filter for cases in which multiple targets move in the $x$–$y$ plane, and false alarms and less-than-unity probability of detection are considered. It is observed that the PHD filter has some unusual behavior that we do not observe in other, well-studied algorithms. In particular, a missed detection can result in loss of the track and consequently a missed detection from one of the sensors can effect other sensors which have a better look to the scenario. Presumably there is a way to fix this, and we are at present pursuing several avenues for this.

References


