Multitarget Tracking with the Probability Hypothesis Density

Group Members
Li Bai, Robert Buckingham, Theresa Hayter, Joushua Strodtbeck, Wei Xiong, Ivan Zorych, Noam Zeev

Mentor
John Hoffman, Lockheed Martin

Abstract
A multitarget tracking problem is studied using the Probability Hypothesis Density (PHD) which is the first-order moment of the posterior density function. The integral of the PHD over a region in state space gives the expected number of targets in that region. Motion and observation models are derived for a set of data provided by Lockheed-Martin. A grid method is used to compute the PHD for this data with two different state spaces: position and position-velocity. Finally, a particle method is used to compute the PHD for a set of simulated data using position as the state space.
1 Introduction

Target tracking is the mathematical art of identifying moving targets given sensor data. Applications include such varied scenarios as traffic control tracking airplanes with radar, fishermen tracking fish with sonar, and security cameras tracking people with infrared. In any case, the data likely contains noise, and identifying the actual targets may be quite difficult.

The mathematical formulation of the single target tracking case is well established. Let \( f_{k|k}(x|z) \) denote the posterior density which tells the probability of finding the target in state \( x \) at \( t = k \) given the first \( l \) observations. Then the idea is to start with \( f_{k|k}(x|z) \), use time prediction to obtain \( f_{k+1|k}(x|z) \), and then Bayes’ rule to get \( f_{k+1|k+1}(x|z) \). Since this is difficult to compute exactly the first-moment vector, or posterior expectation, \( x_{k|k} = \int x f_{k|k}(x|z) \, dx \) is used as an approximation. A constant gain Kalman filter will propagate the first moment in time from \( x_{k|k} \) to \( x_{k+1|k} \) using the predictor step, and to \( x_{k+1|k+1} \) using the update step. For Kalman’s original paper see [6].

For the multi-target case approximation becomes even more important. As Bar-Shalom and Li note, “the tracking effort for \( n \) targets can be substantially more costly than \( n \) times the effort for a single target” [2]. The appropriate first moment of the posterior density is the probability hypothesis density, or PHD. The integral of the PHD gives the expected number of targets in a given region of state space. The PHD is not generally a probability density function because it may integrate to more than 1. The mathematical formulation of the multi-target case has only recently been developed. The PHD was first proposed by Stein and Winter in 1993 [7], and was shown to be the appropriate first-order moment by Mahler in 1997 [4]. The general algorithm is illustrated in the diagram below:

\[
\ldots \rightarrow f_{k|k}(X|Z) \rightarrow f_{k+1|k}(X|Z) \rightarrow f_{k+1|k+1}(X|Z) \rightarrow \ldots
\]

\[
\downarrow \quad \downarrow \quad \downarrow
\]

\[
\ldots \rightarrow D_{k|k} \rightarrow D_{k+1|k} \rightarrow D_{k+1|k+1} \rightarrow \ldots
\]

This work uses the PHD method to analyze a data set provided by Lockheed-Martin. The physical nature of the targets and sensor is unknown, so all assumptions about the motion model and observation model are derived from the data. For each time step, the data gives a vector of intensities at different positions. However, the input to the PHD is a list of observed targets at each time. Therefore it is necessary to turn the vector sensor data into a finite set of observations. The procedure used here will have a tremendous impact on the effectiveness of the PHD.

It is also important to decide what state space the observed targets live in. Any physical system has multiple attributes that may assist in tracking: position, velocity, acceleration, temperature, and so forth. The trick is to pick the most important ones that contain enough information while keeping the problem tractable. The data is analyzed using both position and position plus velocity. Adding a component to state space is seen to increase accuracy but also increase computation time.

The general algorithm for the PHD is to take the vector data, create observations, run the PHD, and finally output target paths.

Sensor data  \rightarrow  Observations  \rightarrow  PHD  \rightarrow  Target paths

Two different implementations of the PHD are presented. The first is a grid method, where the integrals are computed using the points given by the data vector. The second is a Monte Carlo particle method, where points are randomly sampled and weighted appropriately to compute the integral. Particle methods are commonly used for PHD computations. For more information see [8]. However, there are no direct comparisons of particle and grid methods for the PHD in the literature. This work addresses that deficiency.

The outline of the paper is as follows. Section 2 sets up the theoretical framework behind the PHD and introduces the notation. Section 3 introduces the data set and discusses the filtering process used to create a finite set of observations to input into the PHD. Sensor artifacts are also noted. Section 4 details the motion
and observation models. Section 5 presents the results from a grid implementation, and section 6 does the same for a particle method. Section 7 gives a summary and directions for further research.

2 Introduction to General PHD Theory

The theoretical approach to multitarget detection, tracking, and identification is a generalization of the recursive Bayes nonlinear filter. However, even for single-target problems the Bayes nonlinear filter is very computationally challenging. Consequently, it is necessary to develop good approximation strategies for multitargets problems. The approach used here is to track only the first moment of the joint distribution over target states, the probability hypothesis density (PHD). It turns out that the PHD is the best fit approximation of the multitarget posterior. This discussion is taken from Mahler [5], which is a good general reference for the mathematical theory behind the PHD.

2.1 General Set Up

Multitarget state and measurement at time k are modeled as random finite sets (RFS) \( \Xi_k \) and \( \Sigma_k \) on the state and observation spaces \( E_s \) and \( E_o \) respectively. Given a realization \( X_{k-1} \) of the RFS \( \Xi_{k-1} \) at time \( k-1 \), the multitarget state at time \( k \) can be modelled by the RFS

\[
\Xi_k = S_k(X_{k-1}) \bigcup b_k(X_{k-1}),
\]

where \( S_k(X_{k-1}) \) denotes the RFS of targets having survived at time \( k \) and \( b_k(X_{k-1}) \) is the RFS of newly born targets. The statistical behaviour of \( \Xi_k \) is characterised by the conditional “density” \( f_{k|k-1}(X_k|X_{k-1}) \) in an analogous way to the Markov transition density.

Similarly, given a realization \( X_k \) of \( \Xi_k \) at time \( k \), the multitarget measurement is modelled by the RFS

\[
\Sigma_k = \theta(X_k) \bigcup C_k(X_k),
\]

where \( \theta(X_k) \) denotes the RFS of measurements generated by \( X_k \), and \( C_k(X_k) \) denotes the RFS of clutter points. A clutter point is an observation which does not result from a true target.

Now let \( f_{k|k}(X_k|Z^k) \) denote the multitarget posterior density at time step \( k \) conditioned on the time sequence \( Z^k : Z_1, \ldots, Z_k \) of observation sets accumulated at time step \( k \). As for the single target Bayes filter, the following recursive equations hold:

\[
f_{k+1|k}(X|Z^k) = \int f_{k+1|k}(X|W)f_{k|k}(W|Z^k)\delta W
\]

\[
f_{k+1|k+1}(W|Z^{k+1}) = \frac{f_k(Z_{k+1}|x)f_{k+1|k}(X|Z^k)}{\int f_k(Z_{k+1}|X)f_{k+1|k}(X|Z^k)\delta X}
\]

where \( f \ast \delta X \) is the set integral. See [8] for the manipulations of the set integral.

2.2 PHD

When the number of targets is large, this makes it impossible in practice to maintain the joint distribution over target states. This work propagates only the first moment of the joint distribution, the probability hypothesis density (PHD).

Definition 1 (Probability Hypothesis Density) The PHD is the density \( D(x) \) whose integral \( \int_S D(x)dx \) on any region \( S \) is the expected number of targets contained in \( S \).

Assume that the predicted multitarget posterior \( f_{k+1|k}(X|Z^k) \) is approximately Poisson:

\[
f_{k+1|k}(x|Z^k) \approx \exp(-N)D(x)
\]
where $D(x) = D_{k+1|k}(x|Z^k)$ is the PHD of $f_{k+1|k}(x|Z^k)$ and $N = \int D(x)dx$. Let $D_{k|k}$ and $D_{k+1|k}$ be the PHDs of $f_{k+1|k}(X|Z^k)$ and $f_{k+1|k+1}(X|Z(k+1))$ respectively. Then it is shown in [5] that $D_{k+1|k}$ and $D_{k+1|k+1}$ are the best fit approximations of $f_{k+1|k}(X|Z^k)$ and $f_{k+1|k+1}(X|Z^{k+1})$, and the predictor $D_{k|k} \rightarrow D_{k+1|k}$ and the corrector $D_{k+1|k} \rightarrow D_{k+1|k+1}$ will be lossless.

Normally, the multitarget posteriors $f_{k|k}(X|Z^k)$ and the likelihood function $f_{k+1|k}(Z|X)$ are not Poisson distributed. But if the covariance and clutter densities are small then observations will be densely accumulated around target states, confusion due to clutter will be small, hence the multitarget posteriors will be roughly characterized by Poisson distribution.

Finally, the PHD satisfies the following recursive formulas:

\begin{align}
D_{k+1|k} &= b_{k+1|k}(x) + \int (p_d(w)f_{k+1|k}(x|w))dw \\
D_{k+1|k+1} &= \sum_{z \in Z} \frac{p_o(z)f_{k+1}(z|x)}{\lambda c(z) + \int p_d(x)f_{k+1}(z|x)D_{k+1|k}(x)dx} + 1 - p_d(x)
\end{align}

where $f_{k+1|k}(y|x)$ is single target Markov transition density, $p_d(x)$ is the probability one target will survive from time step $k$ to time step $k+1$, and $b_{k+1|k}(y)$ is the birth probability of new targets from time step $k$ to time step $k+1$.

Figure 1: Original data from Lockheed Martin. The horizontal axis is $x_n$ and the vertical axis is $k$, with $k = 1$ at the top. Brighter points represent larger values.
3 Pre-filtering and Analysis

3.1 Pre-filter

The data is in the form of a 1200x845 matrix $A$ where $A_{k,j}$ is the intensity returned by the sensor at the $j^{th}$ horizontal position and the $k^{th}$ time step. See figure (1). However, since the PHD operates on $\mathcal{F}(\mathbb{R}^n)$, it is necessary to extract finite sets of observations from the rows of $A$. This step is referred to as pre-filtering. This requires finding a simple and computationally fast way to remove as much of the noise as possible while still retaining the paths.

Outside of certain problem areas (i.e. $450 < j < 550$) the noise appears to be distributed uniformly. A histogram was created from a 301x86 section of the data that consisted almost entirely of white noise $A_{T,J} = [350,650], J = [35,120]$ (Figure 2)). The noise appears to fit a gamma distribution

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} y^{\alpha-1}e^{-y/\beta}$$

with $\alpha = 2.86$ and $\beta = 1.29 \times 10^3$.

![White Noise Distribution](image)

Figure 2: Histogram of the White Noise

By this approximation, 94% of the noise has brightness level below 7500, and 99% of it lays below level 11000. By visual inspection, all of the targets lay above level 7500, and most of them lay above 11000. This suggests a simple pre-filter: All $A_{k,j} > 11000$ were retained, and $7500 < A_{k,j} < 11000$ were retained only if $A_{k-1,j+1} > 7500$. All other values were set to zero. Since it was assumed that the positions of any stationary targets were known beforehand, these tolerances were changed to $3 \times 10^4$ and $2 \times 10^4$ for $450 \leq j \leq 550$ due to the massively increased noise brightness and density around the intense stationary target near the middle of the data. Finally, all peak brightnesses were extracted from the data to give the observations (figure (3)).
3.2 Noise Distribution

The remaining noise appears to be especially distributed around the stationary targets. However, around bright \((A_{k,j} > 1.5 \times 10^5)\), fast-moving targets, there are especially large bursts of very bright noise (figure (4)). While almost all the noise lies below 11000, the noise around such targets has intensity between \(3 \times 10^4\) and \(9 \times 10^4\). Since the basic pre-filtering algorithm has no way to account for such anomalies, to estimate the noise position density function \(c(x)\), a histogram was taken only of the non-anomalous noise and a curve was approximated (figure (6)). The expected number of clutter points \(\lambda\) was estimated by subtracting the approximate number of targets (30) from the total number of observations divided by 1200, the number of timesteps. A more accurate estimate can be made by taking into account brightness and velocity. Since the signal to noise ratio is so low, velocity is very difficult to estimate. However, expected velocities and positions can be drawn from \(D_{k+1|k}\) and used to estimate both \(\lambda\) and \(c(x)\).

3.3 Sensor Anomalies

The filtered data appears to contain artifacts created by the actual sensor. In figure(3.3) below, the red arrow indicates an actual stationary target while the yellow arrows indicate what is believed to be reflections or ghost targets of the real target. These real and ghost targets are positioned exactly 120 pixels apart. Likewise, moving targets also create ghost targets. Near the real stationary target are four colored arrows pointing out real moving targets. The other arrows correspond to one of the respective ghost targets. In the filtered data, the only portion of the real target that the sensor creates ghosts for is of an intensity or brightness of \(10^5\). The corresponding ghosts decrease in intensity as they get further away from the actual target.

Notice that the sensor produces 7 and 6 ghost targets for the green and orange real targets, respectively. The actual position distribution of the green real and first four ghost targets can be seen in table(3.3) below. On the other hand, the blue and pink ghost targets can only be seen near stationary targets on either side.
of the real stationary target. This is probably solely due to the filter used and can not be verified due to the difficulty of distinguishing targets from clutter in the unfiltered data.

Information obtained from the ghost targets is captured in the observation model and imputed into the Probability Hypothesis Density function. The PHD then propagates this information forward and the output is that a target is detected when in actuality an anomaly due to the sensor. Obviously, this phenomenon creates a problem for the PHD. To circumvent this problem, either a way to filter these ghosts needs to be added to the observation model or modifications need to be made to the sensor.

<table>
<thead>
<tr>
<th>Target</th>
<th>Position at $k \approx 627$</th>
<th>Position of Stationary Reference Target</th>
<th>Position at $k \approx 722$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT 1</td>
<td>62</td>
<td>22-23</td>
<td>off screen</td>
</tr>
<tr>
<td>GT 2</td>
<td>179</td>
<td>142-143</td>
<td>off screen</td>
</tr>
<tr>
<td>GT 3</td>
<td>303</td>
<td>262-263</td>
<td>91</td>
</tr>
<tr>
<td>GT 4</td>
<td>421</td>
<td>382-383</td>
<td>213</td>
</tr>
<tr>
<td>RT 5</td>
<td>541</td>
<td>502-503</td>
<td>333</td>
</tr>
</tbody>
</table>

Targets are labelled from left to right in increasing order with GT = Ghost Target and RT = Real Target. $k$ refers to the time step in the data.
4 The Motion and Observation Models

Two different state space models were implemented for the PHD. The first uses only position, and the second uses position and velocity. A natural extension would be to include intensity. For each it is necessary to specify the motion model (how targets move) and the observation model (how the sensor works).

4.1 The motion model for the position PHD

When position is the only component of the state space, the state is given simply by \( x = z \). The motion model encodes how the state of the targets is expected to change from time \( k \) to \( k + 1 \). This includes the birth model \( b_{k+1|k}(x) \), the survival model \( p_s(x) = 0.99 \), and the single-target Markov transition density \( f_{k+1|k}(x|y) \). These are the terms used to compute the prediction PHD, \( D_{k+1|k}(x) \).

The high level of noise, and lack of knowledge of the physical nature of the targets, makes it difficult to detect any births with reasonable certainty. Therefore the birthrate is assumed to be zero, so that

\[
b_{k+1|k}(x) = 0.
\]

(9)

The probability of one target spawning a new target is also assumed to be zero.

Almost all targets survive for at least a hundred time steps, with many persisting through the entire data set. The probability of survival is taken to be

\[
p_s(x) = 0.99.
\]

(10)

Noise makes it difficult to tell whether any targets actually disappear, so this estimate may be low.

If a target has position \( x_k \) at \( t = k \), then at \( t = k + 1 \) the position of the target is assumed to be normally distributed with mean \( x_k \) and prediction standard deviation \( \sigma_p \). That is,

\[
x_{k+1} = x_k + N(x_k, \sigma_p).
\]

(11)

\( f_{k+1|k}(x|y) \) gives the probability of a target moving from state \( y \) at \( t = k \) to state \( x \) at \( t = k + 1 \), given all the information contained in the first \( k \) observations. Assumption (11) means that \( f_{k+1|k}(x|y) \) is a normal distribution with mean \( y_n \), and standard deviation \( \sigma_p \),

\[
f_{k+1|k}(x|y) = \frac{1}{\sigma_p \sqrt{2\pi}} e^{-\frac{(x_n-y_n)^2}{2\sigma_p^2}}.
\]

(12)

If a target is located at position \( y_n \) at \( t = k \), then at \( t = k + 1 \) it is expected to be located in \([y_n - 2\sigma_p, y_n + 2\sigma_p]\) about 95 percent of the time.

4.2 The observation model for the position PHD

The observation model details the accuracy of the sensor. This information is represented by the probability of detection \( p_D(x) \), the single-target likelihood density \( f_{k+1}(z|x) \), the expected number of clutter points \( \lambda \), and the clutter distribution \( c(z) \). These are the terms used to compute the update PHD, \( D_{k+1|k+1}(x) \).

The probability of detection can be estimated by following target trajectories and counting the number of missed observations. There is no discernable difference in the number of targets detected at different position values, so for this model

\[
p_D(x) = 0.95.
\]

(13)

Since it is unknown how many targets are completely missed by the detector, this number may be high. Also, due to the filtering method used, \( p_D \) depends on intensity. For example, the probability of detecting a target with intensity less than 7500 is zero. Therefore, an observation model including intensity would need to use a modified \( p_D \).
The single-target likelihood density tells the probability that an observed target with true state $x$ will be observed as state $z$. This density is taken to be normal with mean $x_n$ and update standard deviation $\sigma_u = 1$.

$$f_{k+1}(z|x) = \frac{1}{\sigma_u \sqrt{2\pi}} e^{-\frac{(z_n-x_n)^2}{2\sigma_u^2}}$$

(14)

Clutter points are the false observations, where the filter identifies a target that does not actually exist. More noise in the data creates more clutter. The filter returns about 65 to 70 observations at each time step. On the other hand, there are about 25 to 30 observable paths passing through each time step. Therefore the expected number of clutter points at each time step is taken to be

$$\lambda = 40.$$  

(15)

The distribution of clutter varies considerably with different positions. Bright targets, especially the stationary ones, create a higher amount of noise to either side, which gives more clutter points. The clutter distribution was estimated by a normalized sum of weighted Gaussians, each centered on one of the stationary targets. See figure (6). Notice that there is no peak near the stationary target at $x_n = 502$ since the data has been heavily filtered in this area.

4.3 Models for the position-velocity PHD

The second state space model implemented takes into account the velocity of the targets as well as position. The state is represented by $x = (x, v)$. A particle with position $x_k$ and velocity $v_k$ at $t = k$ is assumed to have position $x_k + v_k$ plus a random, uniformly distributed error at $t = k + 1$.

$$x_{k+1} = x_k + v_k + U[-\sigma_p, \sigma_p].$$

(16)

The velocity at $t = k + 1$ is assumed to be normally distributed with mean $v_k$ and standard deviation $\sigma_{p2}$.

$$v_{k+1} = \frac{1}{\sigma_{p2} \sqrt{2\pi}} e^{-\frac{(z_n-x_n)^2}{2\sigma_{p2}^2}}$$

(17)

Therefore

$$f_{k+1}(x|y) = f_{k+1}(x, v|y, w) = \frac{1}{2\sigma_p} \chi_{y+w-\sigma_{p1}, y+w+\sigma_{p1}}N(w, \sigma_{p2}),$$

(18)

where $\chi_{[a, b]}$ is the indicator function on $[a, b]$. The velocity of the targets does not significantly affect the other model parameters, so $b_{k+1}(x), p_S(x), p_D(x), \int_{k+1}(z|x), \lambda$, and $c(z)$ are the same as for the position PHD.
5 The Grid Implementation of the PHD

Two different methods were used to compute the PHD, the grid method and the particle method. The grid method uses all of the points from the discrete sensor data, whereas the particle method uses Monte Carlo sampling.

5.1 Setup for the grid implementation of the position PHD

The first implementation of the PHD uses only position in the state space.

The PHD $D_{0|0}(x)$ is the initial guess for the distribution of targets. There are approximately 25 targets, and it is unknown where they lie a priori. If there is a non-zero birth model, then $D_{1|0}(x)$ can be taken to be zero, and $b_{k+1}(x)$ will detect the initial targets. However, since $b_{k+1}(x) = 0$ in this case, $D_{0|0}(x)$ is taken to be uniform on $[0, 845]$ with integral 25.

$$D_{0|0}(x_i) = \frac{25}{845} \quad i = 1, \ldots, 845$$ (19)

Given $D_{k|k}(x)$, the next step in the algorithm is to compute the corrector $D_{k+1|1}(x)$, which gives the expected position of targets at the first time step given the first $k$ observations.

$$D_{k+1|k}(x) = \int_0^{845} p_s(y)f_{k+1|k}(x|y)D_{k|k}(y)dy$$ (20)

For the grid method this integral is discretized with $\Delta x = 1$. Since $f_{k+1|k}(x|y)$ is a Gaussian, values at $y_j$ sufficiently far away from $x_i$ can be discarded.

$$D_{k+1|k}(x_i) = \sum_{y_j = y_{\text{start}}}^{y_{\text{end}}} 0.99 \frac{1}{\sigma_p \sqrt{2\pi}} e^{-\frac{(x_i-y_j)^2}{2\sigma_p^2}} D_{k|k}(y_n)$$ (21)

Here $y_{\text{start}} = \max(1, x_i - 5\sigma_p)$ and $y_{\text{end}} = \min(x_i + 5\sigma_p, 845)$.

The update PHD $D_{k+1|k+1}(x)$ is computed once the next set of data arrives at $t = k + 1$. Let $Z$ be the set of observed targets at $t = k + 1$.

$$D_{k+1|k+1}(x) = (1 - p_D(x))D_{k+1|k}(x) + \sum_{z \in Z} \lambda c(z) + \int_0^{845} F(y, z)dy$$ (22)

where

$$F(x, z) = p_D(x)f_{k+1}(z|x)D_{k+1|k}(x)$$ (23)

This formula is also discretized for the grid method. Again, since $f_{k+1}(z|x)$ is a Gaussian, only points near $z_i$ are used to compute the integral.

$$D_{k+1|k+1}(x_i) = 0.05D_{k+1|k}(x_i) + \sum_{l=1}^{N} \frac{F(x_i, z_l)}{40c(z_l)} + \sum_{y_j = y_{\text{start}}}^{y_{\text{end}}} F(y_j, z_l)$$ (24)

Here

$$N = \text{number of observations at } t = k + 1$$ (25)

$$y_{\text{start}} = \max(1, z_l - 5\sigma_u)$$ (26)

$$y_{\text{end}} = \min(z_l + 5\sigma_u, 845)$$ (27)

$$F(x_i, z_l) = 0.95 \frac{1}{\sigma_u \sqrt{2\pi}} e^{-\frac{(z_l-x_i)^2}{2\sigma_u^2}} D_{k+1|k}(x_i),$$ (28)

The algorithm then proceeds recursively, finding the prediction PHD $D_{k+2|k+1}(x_i)$, then the update PHD $D_{k+3|k+2}(x_i)$, and so forth.
5.2 Results for the grid implementation of the position PHD

At each time step \( k \), \( \int_a^b D_{k|k}(x)dx \) gives the expected number of targets in the interval \([a, b]\). Figure (7) shows an example of the PHD at \( t = 300 \).

The peaks represent suspected targets. The area under a peak is typically about 0.9. The full results for \( \sigma_p = 1 \) and \( \sigma_p = 2 \) are shown in figures (8) and (9), respectively. Notice that a small value of \( \sigma_p \) such as 1 does not catch many of the quickly moving targets. The PHD filters these targets out as noise since it does not expect points to be moving much from one time step to the next. A larger \( \sigma_p \), such as 2, catches more of the fast targets, but on the downside introduces more noise. This conundrum can be resolved by introducing velocity into the state space.

5.3 Setup for the grid implementation of the position-velocity PHD

Although the general algorithm remains the same, introducing velocity into the state space changes several aspects of the computation. The most significant difference is that the prediction and update PHDs are now functions of two variables, and a double integral is needed to extract the given number of targets.

\( \int_a^b \int_c^d D_{k+1|k}(x,v)dx dv \) gives the expected number of targets at \( t = k + 1 \) with position \( a \leq x \leq b \) and velocity \( c \leq v \leq d \) given the first \( k \) observations. Similarly, \( \int_c^d \int_a^b D_{k+1|k+1}(x,v)dx dv \) gives the expected number of targets at \( t = k + 1 \) with position \( a \leq x \leq b \) and velocity \( c \leq v \leq d \) given the first \( k + 1 \) observations. Also, as noted above, the single-target Markov transition density \( f_{k+1|k}(x|y) \) is different and is now given by equation (18).

Visual inspection of the data shows that very few targets move more than four positions to the left or right in one time step. Therefore the possible velocity values are taken to be \(
\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}
\). The units of velocity are positions moved per timestep.

The PHD is implemented as follows. The initial PHD \( D_{k+1|k+1}(x_0, v_m) \) is still taken to be uniform, but now velocity is taken into account.

\[
D_{k+1|k+1}(x_0, v_m) = \frac{25}{845 \times 9}
\]  \hspace{1cm} (29)

The prediction PHD is given by

\[
D_{k+1|k}(x, v) = \int_{-4}^{4} \int_{0}^{845} p_s f_{k+1|k}(x, v|y, w) D_{k|k}(y, w) dy dw
\]  \hspace{1cm} (30)
Figure 8: The PHD $D_{k|k}(x_t)$ with $\sigma_p = 1$.

Figure 9: The PHD $D_{k|k}(x_t)$ with $\sigma_p = 2$. 
which is discretized as
\[
D_{k+1|k}(x_i, v_m) = \sum_{w_n=-4}^{4} \sum_{y_j=-4}^{4} 0.99 \frac{1}{\sigma_{p1} \sqrt{2\pi}} e^{-(v_m-w_n)^2/2\sigma_{p2}} \frac{1}{2\sigma_{p1}} D_{k|k}(y_j, w_n)
\]
(31)

For the computations below, \(\sigma_{p1} = 1\) and \(\sigma_{p2} = 0.5\).

The formula for the update PHD is
\[
D_{k+1|k+1}(x, v) = (1 - p_D) D_{k+1|k}(x, v) + \sum_{z \in Z} \lambda c(z) + \int_{-4}^{4} \int_{-4}^{4} F(y, w, z) dy
\]
(32)
in which
\[
F(x, z) = p_D f_{k+1}(z|x, v) D_{k+1|k}(x, v).
\]

The discretized version is
\[
D_{k+1|k+1}(x_i, v_m) = 0.05 D_{k+1|k}(x_i, v_m) + \sum_{l=1}^{N} \frac{40c(z_l)}{4} + \sum_{w_n=-4}^{4} \sum_{y_j=-4}^{4} F(y_j, w_n, z_l)
\]
(34)

\(N\), \(y_{\text{start}}\), and \(y_{\text{end}}\) are given by equations (25)-(27), and
\[
F(x_i, v_m, z_l) = 0.95 \frac{1}{\sigma_u \sqrt{2\pi}} e^{-(z_l-x_i)^2/2\sigma_u^2} D_{k+1|k}(x_i, v_m).
\]
(35)

For the computations, \(\sigma_u\) is taken to be 1.

### 5.4 Results for the grid implementation of the position-velocity PHD

Velocity is introduced into the state space to track the quickly moving targets without introducing more noise. The added difficulty is that \(D_{k|k}(x, v)\) now depends on three parameters: position, velocity, and time. To visualize the results, pick a specific \(v = v_0\) and look at the plot of \(D_{k|k}(x, v_0)\) as a function of position and time, as before.

Figure (10) shows the PHD for \(v_0 = -3\). Notice that the targets moving with negative velocity have the highest density, and the stationary targets have lower density. Targets moving to the right are not even detectable. Figure (11) shows the PHD for \(v_0 = 0\). Here the stationary targets have the highest probability. Finally, figure (12) shows the PHD for \(v_0 = 3\). Targets moving to the right have high density, and targets moving to the left are not discernable. Finally, figure (13) shows \(\sum_{v_m=-4}^{4} D_{k+1|k+1}(x_i, v_m)\), which gives the density of targets irrespective of velocity. Comparing figures (9) and (13) shows that including velocity in the state space with \((\sigma_{p1}, \sigma_{p2}, \sigma_u) = (1, 0.5, 1)\) picks up several more moving targets than the position-only PHD with \((\sigma_p, \sigma_u) = (2, 1)\) with about the same amount of clutter. To pick up that many moving targets with the position-only state space requires increasing \(\sigma_p\), which introduces significantly more clutter.

The disadvantage to using velocity in the state space is that the computation of \(D_{k|k}(x)\) for 1200 time steps goes from taking approximately two minutes to approximately two hours. Nevertheless, this rate is still sufficiently high that, depending on the sensor rate and processing speed available, a real-time tracking algorithm is certainly feasible.
Figure 10: The PHD $D_{k|k}(x_t, v_m = -3)$.

Figure 11: The PHD $D_{k|k}(x_t, v_m = 0)$. 
Figure 12: The PHD $D_{k|k}(x_t, v_m \equiv 3)$.

Figure 13: $\sum_{v_m = -4}^{4} D_{k|k}(x_t, v_m)$. 
6 Particle filter

The particle realization of the PHD is calculated using the Bootstrap filter approach. The main idea of the Bootstrap filter is to approximate the posterior distribution with distribution of weighted particles. Here we use the Bootstrap filter for the following is implemented for the following nonlinear problem: the state vector, $x_k$, evolves according to the system model $x_{k+1} = f_k(x_k, w_k)$, where $w_k$ has a known PDF function. We can measure the observations $z_k$ over the state vector: $z_k = h_k(x_k, v_k)$ where the PDF function of $v_k$ is also known.

The posterior density is recursively calculated $p(x_k | Z^{(k)})$ as follows:

$$p(x_k | Z^{(k-1)}) = \int p(x_k | x_{k-1})p(x_{k-1} | Z^{(k-1)})dx_{k-1} \quad (Prediction)$$

$$p(x_k | Z^{(k)}) = \frac{p(z_k | x_k)p(x_k | Z^{(k)})}{\int p(z_k | x_k)p(x_k | Z^{(k-1)})dx_k} \quad (Update)$$

**Particle implementation**: assume a set of random samples $\{x_k(i)\}$ is sampled from PDF $p(x_{k-1} | Z^{(k-1)})$. The bootstrap algorithm propagates these particles in such a way that the result of the iteration of the algorithm is a set of $\{x_k(i)\}$ which are approximately distributed as $p(x_k | Z^{(k)})$.

**Prediction**: Each sample is passed through the system model according to the motion model. The result is the set of particles $\{x_k^*(i)\}$.

**Update**: Using the measurement $z_k$ evaluate weights $q_i = \frac{p(z_k | x_k^*(i))}{\sum_p p(z_k | x_k^*(i))}$ of every particle and resample. After that go to prediction step again.

This method may be applied to implement the recursive estimation of the PHD. See [1],[3],[8] for details.

**Numerical Results**: Particle implementation of PHD has tremendous advantage comparing to the grid method, it runs many times faster and allows the online estimations of multitarget states.

Simulated sets of data were created with an average of six targets and low noise (figure (14)). Both the grid method (figure (15)) and the particle method (figure (16)) were used to detect targets. In both cases the state space was position only. Targets were completely reconstructed by the particle method, and completely reconstructed after a certain amount of time by the grid method. This wait could be addressed by increasing the expected number of births. The computation time for both methods was similar. The particle filter as implemented does not have a clutter model, which is evident from figure (16).

7 Conclusion and Further Directions

The Probability Hypothesis Density was applied to a multi-target tracking problem derived from a set of noisy sensor data provided by Lockheed-Martin. Since the PHD assumes observations are in the format of finite sets, it was necessary to transform the vectors of sensor data into sets of observations. After cutting off all the noise below a certain level and extracting contiguous sequences of points above another smaller level, observations were identified by extracting the positions of intensity peaks on each vector. This pre-filtering identified approximately 25 "actual" targets (some were actually sensor ghosts, but were treated like targets for purposes of this project) and 40 clutter points at each time step. The PHD was computed using a grid method which discretizes the integrals over state space using a regular grid. Two different state spaces were used, position and position-velocity. Adding velocity to the state space resulted in the detection of more quickly targets that were previously missed without adding more clutter. However, the computation time was increased by an order of magnitude.

A method more popular than the grid method is the particle method. Particle methods sample the domain randomly and weight these points appropriately. A particle method was implemented for simulated noisy data.
Figure 14: Simulated data.

Figure 15: Results of the grid method for the simulated data.
using position as the only state space variable. When compared to the grid method, both did a good job of detecting targets and had similar computation times.

There are several areas for future work. The clutter model would be enhanced by increasing clutter near quickly moving targets. The grid method could be improved by adding intensity to the state space, although this will likely be computationally prohibitively expensive. This would enhance the PHD by allowing a dynamic clutter model expecting more clutter around high intensity targets, a smarter detection model, and a more robust survival model. The particle method could be enhanced by adding velocity or brightness to the state space. The final step where target paths are extracted from the PHD also needs to be addressed. Finally, it would be useful to directly accept vector sensor data, thereby bypassing the entire pre-filtering stage.
References


