A Look at the PMHT

David F. Crouse†, Marco Guerriero†, Peter Willett†,
Roy Streit‡, and Darin Dunham∗
†Dept. of Electr. and Comp. Engineering, University of Connecticut, Storrs, CT 06269, USA
‡Metron Inc., 1818 Library Street, Reston, VA 20190, USA
∗Vectraxx Inc., 12131 Howards Mill Road, Glen Allen, VA 23059, USA

Abstract – We combine concepts from numerous papers to provide a derivation and description of a generalized Probabilistic Multi-Hypothesis Tracker (PMHT) that can track multiple targets in a cluttered environment, utilizing multiple sensors and feature measurements, if available. We also provide a simplified method of performing the maximization step of the algorithm when multiple sensors are used, a consistent covariance approximation of the algorithm when using multiple sensors, and we explore the use of deterministic annealing to improve performance and discuss implementation difficulties. Additionally, we derive the complexity of the PMHT and the JPDAF to better understand the advantages of each algorithm.

1 Overview

Since its creation by Streit and Luginbuhl in 1993 [1], much research has been done on the Probabilistic Multi-Hypothesis Tracker (PMHT). In this paper, we combine the concepts from past works and provide a general version of the PMHT algorithm allowing for tracking in the presence of clutter and missed detections and the utilization of classification data, range rate information and multiple sensors. This version makes no changes to the basis of the original algorithm, which is the Expectation Maximization (EM) algorithm. As a result, this generalized PMHT algorithm may be used as an improved foundation for other PMHT-based algorithms, such as the Multi-Frame Assignment PMHT (MF-PMHT) accounting for missed detections by Blanding, Willett, Streit, and Dunham [2].

In the subsequent sections we derive the general form of the PMHT algorithm. The algorithm is summarized in section 7. Section 2 describes the EM algorithm, which forms the basis of the PMHT. Section 3 derives the state estimates within the PMHT, discussing implementation issues associated with precision problems in 3.1, how and why one might wish to include deterministic annealing to improve performance in 3.1. Section 5 then discusses state covariance estimation in the PMHT. In section 4 we compare the complexity of the PMHT against that of the Joint Probability Data Association Filter (JPDAF), which is a popular non-batch tracking algorithm1. Section 6 provides a simulation of the PMHT with multiple sensors demonstrating that deterministic annealing can significantly improve tracker performance when using multiple sensors. Section 8 summarizes the paper. This paper is an abbreviation of [3], which has more extensive proofs and a detailed history of the algorithm.

2 The Expectation Maximization Algorithm

The PMHT is based on the Kalman filter (see, e.g., [4]) and the EM algorithm. The EM Algorithm, discovered by Dempster, Laird and Rubin [5], is a method of determining the maximum likelihood (ML) or maximum a posteriori (MAP) estimate of data given incomplete information. The algorithm is extensively covered in the monograph by McLachlan and Krishnen [6].

Let \( \mathbf{X} \) be an unknown random quantity whose MAP estimate we would like to find. Let \( \mathbf{Z} \) be the set of observations which are dependent upon \( \mathbf{X} \) and a set of unobservable random variables \( \mathbf{K} \). We would like to find the MAP estimate of \( \mathbf{X} \) without having to determine \( \mathbf{K} \), which might be a difficult or computationally complex task. The MAP estimate of \( \mathbf{X} \) may be expressed as follows:

\[
\hat{\mathbf{X}}_{MAP} = \arg \max_{\mathbf{X}} E \{ \log p(\mathbf{X}|\mathbf{Z}) \} \tag{1}
\]

in which \( p \) in (1) represents a probability density function (PDF). The expectation comes from the Law of Total Probability eliminating the unobservable random variable \( \mathbf{K} \). However, in many cases the expectation may be difficult to evaluate. The EM algorithm avoids direct computation of this expectation. Define the following function:

\[
Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) = \int_{\mathbf{K}} \log \left( p(\mathbf{Z}, \mathbf{X}^{(n+1)}|\mathbf{K}) \right) p(\mathbf{K}|\mathbf{X}^{(n)}, \mathbf{Z}) \, d\mathbf{K} \tag{2}
\]

The integration in (2) is defined over whichever measure is appropriate for \( \mathbf{K} \), which may be discrete. The EM algorithm is as follows: in each step, \( \mathbf{X}^{(n+1)} \) is found as:

\[
\mathbf{X}^{(n+1)} = \arg \max_{\mathbf{X}^{(n+1)}} Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) \tag{3}
\]

1It is also referred to as the Multi-Sensor JPDAF (MSJPDAF) in the multisensor case.
n is then incremented and one continues until a desired level of convergence has been attained.

In order to improve convergence to the global maximum, the Deterministic Annealing (DA) EM algorithm was developed by Ueda and Nakano [7]. This was applied to the PMHT first in 1999 by Strandlie and Zerubia [8] and was later applied in a more general form by Wieneke and Koch [9]. When tracking a single target, the basic PMHT algorithm with deterministic annealing is identical to the Deterministic Annealing Filter by Frühwirth and Strandlie [10].

3 The PMHT Algorithm: State Estimates

We shall now derive a general form of the PMHT algorithm allowing for the presence of clutter, multiple sensors and the use of classification information. The most general form of the PMHT is based on the basic Kalman filter, although variants based on other models have been considered [11].

Given M targets, the state vector at time t for the mth target shall be designated as \( x_m(t) \). The observation originating from the mth target shall be designated \( y_m(t) \). The basic discrete-time kinematic motion and observations equations are as follows:

\[
x_m(t+1) = F_m(t)x_m(t) + v_m(t) \\
y_m(t) = H_m(t)x_m(t) + w_m(t)
\]

\( v_m(t) \) and \( w_m(t) \) are assumed to be uncorrelated and distributed Gaussian with zero mean and covariances \( Q_m(t) \) and \( R_m(t) \). The covariance of the true measurement from the mth target, \( R_m(t) \) describing \( w_m(t) \) from (5), corresponds to the covariance of the “correct” measurement out of the set of all measurements at time t, whereby \( R_{r,s}(t) \) shall represent the covariance of measurement \( r \) from sensor \( s \) based upon the location of the observation, without stating a particular associated target.

Let \( Z \) be all of the measurements and classification information from time \( t = 1 \) to \( N \). Let \( X \) be the states of all of the targets over the same time period and \( K \) be the set of associations between targets and measurements. Let there be a total of \( S \) sensors that take measurements synchronously. If \( z_{r,s}(t) \) is the measurement \( r \) at time \( t \) from sensor \( s \) came from target \( m \), then we shall denote said association by \( k_{r,s}(t) = m \). We would like to use the EM algorithm to estimate \( X \) without explicitly determining which set of \( k_{r,s}(t) \) from the set of all possible target to measurement associations, \( K \), is correct. We shall consider clutter to be target \( m = 0 \).

The inclusion of classification measurements in the PMHT was first discussed by Davey, Gray and Streit [12]. We shall define \( z_{r,s}^C(t) \) to be the classification data associated with measurement \( r \) from sensor \( s \) at time \( t \). Including classification in the PMHT means estimating the type of each target. This is done via a confusion matrix, \( C \), whose elements are defined as follows:

\[
c(i,m) = \Pr(z_{r,s}^C(t) = i | k_{r,s}(t) = m) \quad (6)
\]

\( i \) in (6) represents the i\(^{th}\) classification out of the set of all \( M_C \) possible classifications. The true classification of each target is assumed to be time-invariant and independent of the state, which is why \( c_{i,m} \) is not indexed against time. The confusion matrix is the estimated probability that a target or a clutter measurement has a certain associated appearance. The confusion matrix shall be estimated along with \( X \) in the PMHT algorithm.

Let us find the first PDF in (2), including \( C \) next to \( X \) as an unknown to be estimated. Let \( n_{i,s}(t) \) be the number of measurements at time \( t \) that came from sensor \( s \). In order for \( p(Z, X, K, C) \) to be written, \( n_{i,s}(t) \) shall be conditioned on \( n_{i,s}(s) \). However, we will not explicitly write this conditioning except when necessary. \( p(Z, X, K, C) \) is given as follows:

\[
p(Z, X, K, C) = p(X)p(Z, K, C | X)
\]

\[
e \left( \prod_{m=1}^{M} \prod_{t=2}^{T} \prod_{i=1}^{T} \prod_{t=1}^{S} \prod_{s=1}^{n_{i,s}(s)} \Pr(k_{r,s}(t) | x_{k_{r,s}(t)}(t), n_i) \right)
\]

\[
\times p(z_{r,s}(t) | k_{r,s}(t), x_{k_{r,s}(t)}(t))
\]

\[
\times p(z_{r,s}^C(t) | k_{r,s}(t), x_{k_{r,s}(t)}(t))
\]

\[
\times \epsilon(z_{r,s}^C(t), k_{r,s}(t))
\]

\[
= \left( \prod_{m=1}^{M} \prod_{t=2}^{T} \prod_{i=1}^{T} \prod_{t=1}^{S} \prod_{s=1}^{n_{i,s}(s)} \Pr(k_{r,s}(t) | x_{k_{r,s}(t)}(t), n_i) \right)
\]

\[
\times p(z_{r,s}(t) | k_{r,s}(t), x_{k_{r,s}(t)}(t))
\]

\[
\times p(z_{r,s}^C(t) | k_{r,s}(t), x_{k_{r,s}(t)}(t))
\]

\[
\times \epsilon(z_{r,s}^C(t), k_{r,s}(t))
\]

In (7) the PDF \( p(z_{r,s}(t) | k_{r,s}(t), x_{k_{r,s}(t)}(t)) \) depends upon whether the measurement came from clutter or from a target and is given as follows:

\[
p(z_{r,s}(t) | k_{r,s}(t), x_{k_{r,s}(t)}(t)) =
\]

\[
\begin{cases}
\mu(t, z_{r,s}(t)) & \text{if } k_{r,s}(t) = 0 \\
N(z_{r,s}(t); \hat{y}_{k_{r,s}(t)}(t), R_{r,s}(t)) & \text{if } k_{r,s}(t) \neq 0.
\end{cases}
\]

In (8), \( \mu \) denoted the PDF of the clutter, which we shall assume to be continuous as a function of \( z_{r,s}(t) \) and \( \hat{y}_{k_{r,s}(t)}(t) \), the estimate of \( y \) from (5). That is,

\[
\hat{y}_{k_{r,s}(t)}(t) = H_{k_{r,s}(t)}(t)x_{k_{r,s}(t)}(t)
\]

Define \( w_{k_{r,s}(t), r}(t, s) \) to be the probability of a particular measurement-target assignment at time \( t \) conditioned on \( x_{k_{r,s}(t)}(t), Z(t) \) and \( C, n_{i,t} \), whereby clutter is target \( k_{r,s}(t) = 0 \). One instance of \( K \) defines \( k_{r,s}(t) \) over all measurements, \( r \), and sensors \( s \), for all time in the batch. The sum over \( K \) is equal to the sum over all sensors of the sum
over $K_s$, which is defined as follows:

$$
\sum_{K_s} (\bullet) = \sum_{k_{1,s}(1) = 0}^{M} \sum_{k_{2,s}(1) = 0}^{M} \ldots \sum_{k_{n_s (s), s}(T) = 0}^{M} (\bullet) \tag{10}
$$

Under the basic PMHT assumption, because each target can produce more than one measurement, all of the values of $w_{k_{r,s}(t),r}(t, s)$ at a particular time are independent. Additionally, because the current state and observation set are given, the values of $w_{k_{r,s}(t),r}(t, s)$ are also independent as a function of time. Because of this independence, $p(\mathbf{K}|\mathbf{X}, \mathbf{C}, \mathbf{Z})$, the second PDF in (2), may be obtained directly by multiplying the marginal probabilities over all time and measurements for all of the assignments:

$$
p(\mathbf{K}|\mathbf{X}, \mathbf{C}, \mathbf{Z}) = \prod_{s=1}^{S} \prod_{t=1}^{T} \prod_{r=1}^{n_t} w_{k_{r,s}(t),r}(t, s) \tag{11}
$$

We would now like to determine the measurement target assignment probability $w_{k_{r,s}(t),r}(t, s)$. A full derivation is given in [3] and is a natural extension of that given in other papers, such as [13]. Define $\pi_{k_{r,s}(t)}(n_t(s), t) \triangleq \Pr (k_{r,s}(t) = m|\mathbf{x}_r(t), n_t(s))$. The $\pi_{k_{r,s}(t)}(n_t(s), t)$ values were derived using “imaginary” targets by Wieneke and Koch [9]. They were rederived in a simpler form in [3] and the solution assuming that all targets have same probability of detection $D_k(s)$ when viewed by sensor $s$, and letting $\xi(k)$ be the probability that there are $k$ clutter points at time $t$, from sensor $s$ is given in (13)

$$
\sum_{k_{1,s}(1) = 0}^{M} \sum_{k_{2,s}(2) = 0}^{M} \ldots \sum_{k_{n_s (s), s}(T) = 0}^{M} (\bullet)
$$

For the case where $k_{r,s}(t) \neq 0$, that is when measurement $r$ from sensor $s$ is not clutter, then $w_{k_{r,s}(t),r}(t, s)$ is given in equation (12) (in the final solution to the PMHT algorithm, it will turn out that one never needs to evaluate $w_{0,r}(t, s)$).

Combining (11) and (7b) and omitting the constant $p(\mathbf{Z})$ we may form the Q function for the basic EM algorithm.

The superscripts in parentheses in (14b) indicate whether the values in question are to be calculated using the current or the previous estimate of $\mathbf{X}$ in the EM algorithm. Using identities pointed out by Davey [14] in his thesis, (14b) may be simplified as given in equation (15).

The maximization of the state gradient, $\nabla_{\mathbf{X}_{m+1}} Q_{\mathbf{X}}$ from (15), is performed indirectly by finding an equation with the same derivative and thus the same inflection points. One such function is given in (16). The synthetic measurements $\mathbf{z}_{m,s}(t)$ and $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ from (16) are defined as follows:

$$
\tilde{\mathbf{R}}_{m,s}(t)^{-1} = \sum_{r=1}^{n_t(s)} w_{m,r}(t) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \tag{17}
$$

$$
\mathbf{z}_{m,s}(t) = \mathbf{R}_{m,s}(t) \left( \sum_{r=1}^{n_t(s)} w_{m,r}(t) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \right) \tag{18}
$$

Equation (16) is the joint likelihood function of $M$ targets for which there are observations from multiple sensors but no data observation uncertainty. The second term of (16) may be rewritten as follows:

$$
-\frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} (\mathbf{z}_m - \mathbf{H}_m \mathbf{x}_m^{(n+1)}(t))' \tilde{\mathbf{R}}_m(t)^{-1} (\mathbf{z}_m - \mathbf{H}_m \mathbf{x}_m^{(n+1)}(t)) \tag{19}
$$

Letting $\mathbf{I}_{m,s}$ be an identity matrix whose width is equal to the number of states in $\mathbf{x}_m$, $\mathbf{z}_m(t)$ and $\tilde{\mathbf{R}}_m(t)$ are given as follows:

$$
\tilde{\mathbf{z}}_m(t) = [\tilde{\mathbf{z}}_{m,1}(t), \tilde{\mathbf{z}}_{m,2}(t), \ldots, \tilde{\mathbf{z}}_{m,S}(t)] \tag{20}
$$

$$
\mathbf{H}_m = [\mathbf{I}_{m,1}, \mathbf{I}_{m,2}, \ldots, \mathbf{I}_{m,S}] \tag{21}
$$

$$
\mathbf{R}_m(t) = \text{diag} [\tilde{\mathbf{R}}_{m,1}, \tilde{\mathbf{R}}_{m,2}, \ldots, \tilde{\mathbf{R}}_{m,S}] \tag{22}
$$

Substituting equation (19) into (16) is equivalent to a single-sensor system with no data association uncertainty having measurements determined given by (20), (21) and (22). The maximization of $\hat{Q}$ is thus the maximization of a single sensor system, the solution of which is well known (e.g., [4]) to be the use of the Kalman smoother. This is a simpler approach than using the Levenberg-Marquardt nonlinear regression procedure, as suggested by Giannopoulos, Streit and Swaszek [15] in the original derivation of the PMHT with multiple sensors.

Gan and Harris [16] showed that if at a particular time for a particular track all sensors have the same measurement matrix , which being $\mathbf{I}_{m,1}$ is true in this case, then the above method of merging the states is equivalent to a simpler method. Namely, each track is updated using a single, merged measurement given as follows:

$$
\mathbf{z}_m(t) = \mathbf{R}_m(t) \sum_{s=1}^{S} \tilde{\mathbf{R}}_{m,s}(t)^{-1} \mathbf{z}_{m,s}(t) \tag{23}
$$

$$
\mathbf{R}_m(t) = \left( \sum_{s=1}^{S} \tilde{\mathbf{R}}_{m,s}(t)^{-1} \right)^{-1} \tag{24}
$$

In general, except when range-rate information is provided by the sensors, all $\mathbf{H}_{r,s}$ for a particular sensor, $s$, will be the same for all measurements. In this instance, as men-
\[ w_{k_r,s(t),r}(t,s) = \frac{\pi_{k_r,s}(t)(n_t(s),t)N \{ z_{r,s}(t); \hat{y}_{k_r,s}(t) \}, R_{r,s}(t)}{\pi_0(n_t(s),t)\mu(t,z_{r,s}(t))c(z_{r,s}(t),0) + \sum_{m=1}^{M} \pi_m(n_t(s),t)N \{ z_{r,s}(t); \hat{y}_m(t), R_{r,s}(t) \} c(z_{r,s}(t),m)} \] (12)

\[ \pi_m(n) = \begin{cases} 1 - M\pi_1(n) & m = 0 \\ -\frac{2F_0}{2F_0} \left[ 1 - M, 1 - n; -\frac{P_D}{(1-P_D)\lambda V} \right] & m \neq 0 \end{cases} \] (13)

\[ Q(X^{n+1}, C^{n+1}, X^n, C^n) = \sum_\mathbf{K} \log \left( p \left( Z, X^{(n+1)}, C^{(n+1)}, \mathbf{K} \right) \right) p(\mathbf{K} | X^n, C^n, Z) \] (14a)

\[ = \log \left( \prod_{m=1}^{M} p \left( x^{(n+1)}_m(1) \Big\| x^{(n+1)}_m(t_x) \right) \right) \prod_{t_x=2}^{T} \prod_{t_x=2}^{T} p \left( x^{(n+1)}_m(t_x) | x^{(n+1)}_m(t_x - 1) \right) \]

\[ + \sum_{S} \sum_{T} n_x(s) \log \left( \pi_{k_r,s}(1)(n_t(s),t) \right) \prod_{s=1}^{S} \prod_{t=1}^{T} \prod_{r=1}^{r_1} w_{k_{r_1},s}(t_1,s_1) \]

\[ + \sum_{S} \sum_{T} n_x(s) \log \left( c(z_{r,s}(t), k_{r,s}(t)) \right) \prod_{s=1}^{S} \prod_{t=1}^{T} \prod_{r=1}^{r_1} w_{k_{r_1},s}(t_1,s_1) \]

\[ + \sum_{S} \sum_{T} n_x(s) \log \left( c(z_{r,s}(t), m) \right) w_{m,r}(t,s) \] (14b)

\[ Q(X^{n+1}; X^n) = \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{r=1}^{r_1} \sum_{m=1}^{M} \log \left( \pi_m(n_t(s),t) \right) w_{m,r}(t,s) \left\{ Q_{\Pi} \right\} \]

\[ + \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{r=1}^{r_1} \sum_{m=1}^{M} \log \left( c(z_{r,s}(t),m) \right) \left\{ Q_{\mathbf{C}} \right\} \]

\[ + \log \left( \prod_{m=1}^{M} p \left( x^{(n+1)}_m(1) \Big\| x^{(n+1)}_m(t_x) \right) \right) \prod_{t_x=2}^{T} \prod_{t_x=2}^{T} p \left( x^{(n+1)}_m(t_x) | x^{(n+1)}_m(t_x - 1) \right) \]

\[ + \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{m=1}^{M} \log \left( N \{ z_{r,s}(t); \hat{y}_m(t), R_{r,s}(t) \} \right) w_{m,r}(t,s) \] (15)

\[ \dot{Q}(X^{n+1}; X^n) = \log \left( \prod_{m=1}^{M} p \left( x^{(n+1)}_m(1) \Big\| x^{(n+1)}_m(t_x) \right) \right) \prod_{t_x=2}^{T} \prod_{t_x=2}^{T} p \left( x^{(n+1)}_m(t_x) | x^{(n+1)}_m(t_x - 1) \right) \]

\[ - \frac{1}{2} \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{m=1}^{M} \left( \bar{z}_{m,s} - x^{(n+1)}_m(t) \right) ' \bar{R}_{m,s}(t)^{-1} \left( \bar{z}_{m,s} - x^{(n+1)}_m(t) \right) \] (16)
tioned in [9], the synthetic measurements may be simplified:
\[  \tilde{z}_{m,s}(t) = \tilde{R}_{m,s}(t) \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) R_{r,s}(t)^{-1} z_{r,s}(t) \]  
(25)
\[  \tilde{R}_{m,s}(t) = \left( \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) R_{r,s}(t)^{-1} \right)^{-1} \]  
(26)

The maximization of the confusion matrix via the gradient, \( \nabla_{C^{n+t}Q_C} \) from (15), is performed under the constraint:
\[  \sum_{i=1}^{MC} c_{i,m} = 1 \]  
(27)

Using (15) and (27), the final solution to the Lagrangian maximization problem gives us the update for the c_{i,m}:
\[  c_{i,m} = \frac{\sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{r=1}^{n_t(s)} \delta (z_{r,s}^{(C)} - i) w_{m,r}^{(n)}(t,s) \sum_{n=1}^{n_t} \sum_{r_1=1}^{n_t} w_{m,r_1}^{(n)}(t_1,s_1)}{\sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) R_{r,s}(t)^{-1}} \]  
(28)

3.1 Using Deterministic Annealing

The use of deterministic annealing can both help the PMHT to converge to the global MAP estimate as well ameliorate precision problems associated with the w’s.

The EM function with deterministic annealing replaces \( p(\mathbf{K} \mid \mathbf{X}^n, \mathbf{Z}) \) in the regular EM algorithm with the following:
\[  p(\mathbf{K} \mid \mathbf{X}^n, \mathbf{C}^n, \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^n, \mathbf{C}^n)^\beta}{\sum_{\mathbf{K}} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^n, \mathbf{C}^n)^\beta} d\mathbf{K} \]  
(29)

The solution for \( p(\mathbf{K} \mid \mathbf{X}^n, \mathbf{Z}) \) given in (11) is a product of \( w \) terms. The addition of the \( \beta \) terms in (29) to (11) can take place without explicitly decomposing \( p(\mathbf{K} \mid \mathbf{X}^n, \mathbf{Z}) \) into the parts listed in (29).

We note that each \( w \) term in (11) has a single value in the numerator as well as a sum of values in the denominator. The inclusion of the \( \beta \) is done by modifying the \( w \) terms as follows. For non-clutter association probabilities, each \( w \) shall be adjusted from (12) as given in equation (30) This is the same as the solution given by Wieneke and Koch [9] without derivation and similar to what Strandlie and Zerubia [8] derived. For precision purposes, the exponentiation of the normal PDFs in (30) is best performed by distributing \( \beta \) to the terms of the normal PDF, rather than evaluating the normal PDF and then exponentiating it. Because \( \beta \leq 1 \), this increases the argument of the exponential function of the PDF, which is where underflow problems are most likely to occur.

4 The Complexity of the JPDAF vs the PMHT

The most complex part of the JPDAF is the evaluation of the measurement assignment probabilities. These are equivalent to the \( w \) values in the single-sensor PMHT, but with slightly different conditioning. The evaluation of these probabilities is complex, because it requires the evaluation and normalization of the likelihoods of all possible target-measurement assignment combinations, a task requiring the evaluation of the exponential function for every likelihood.

In the worst-case scenario, every measurement at time step \( t \) would fall in every target’s gating region. Let \( n_t \) be the number of measurements at step \( t \) and \( M \) be the number of targets. The number of possible target-measurement assignments may be decomposed based upon the number of targets observed and is given in equation (31) The step from (31a) to (31b) was performed by noting that the ratio of the \( a_{l+1} \) and the \( a_l \) term of the sum in (31a) is:
\[  \frac{a_{l+1}}{a_l} = \frac{(1-n_t)(1-M)}{1+l} \]  
(32)

More information on the conversion of sums to hypergeometric functions may be found in [17].

In contrast, although the PMHT allows for more target-measurement assignments than the JPDAF, due to their product form (i.e., the assumed independence of the associations) these do not need to be enumerated individually. The evaluation of each \( w \) term requires evaluating a single normal PDF, and in the end normalizing over all \( w \) terms and a clutter term. Thus, the number of evaluations of the exponential function that need be done for one iteration at one time-step in the PMHT is equal to the number of \( w \) terms, which is \( n_tM \). Thus, if the batch length of the PMHT is \( N \), and \( I \) iterations are used then, noting that the first estimate in the batch does not change with each iteration, the overall complexity of the PMHT is:
\[  A_{PMHT} = IM \sum_{t=2}^{N} n_t \]  
(33)

As shown in table 1, when the number of targets is small, the PMHT will have a higher complexity than the JPDAF. The complexity of the JPDAF scales exponentially with the number of targets, whereas the complexity of the PMHT scales linearly.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( A_{JPDAF} )</th>
<th>( A_{PMHT} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( n_t + 1 )</td>
<td>( 1 \sum_{t=2}^{N} n_t )</td>
</tr>
<tr>
<td>2</td>
<td>( n_t^2 + n_t + 1 )</td>
<td>( 2 \sum_{t=2}^{N} n_t )</td>
</tr>
<tr>
<td>3</td>
<td>( n_t^2 + 2n_t + 1 )</td>
<td>( 3 \sum_{t=2}^{N} n_t )</td>
</tr>
<tr>
<td>4</td>
<td>( n_t^2 - 2n_t^3 + 5n_t^2 + 1 )</td>
<td>( 4 \sum_{t=2}^{N} n_t )</td>
</tr>
</tbody>
</table>

Table 1: The number of combinations considered by the JPDAF versus the PMHT as a function of the number of targets.
$$w_{k_{r,s}(t),r}(t, s) = \frac{(\pi_{k_{r,s}(t)}(n_t(s), t) c (z_{r,s}^C(t), k_{r,s}(t)) \mathcal{N}\{z_{r,s}(t); \tilde{y}_{k_{r,s}(t)}(t), R_{k_{r,s}(t)}(t)\})^2}{(\pi_0(n_t(s), t) \mu(t, z_{r,s}(t)) c (z_{r,s}^C(t), 0))} \times \sum_{m=1}^{M} (\pi_m(n_t(s), t) c (z_{r,s}^C(t), m) \mathcal{N}\{z_{r,s}(t); \tilde{y}_m(t), R_{r,s}(t)\})^2$$ \hspace{1cm} (30)

$$A_{JPDAF} = \sum_{l=0}^{\text{min}(n_t, M)} \{ \begin{array}{c}
\text{Choose which targets are observed} \\
\text{Choose which measurements are observed} \\
\text{Assign the measurements to the targets}
\end{array} \} \hspace{1cm} (31a)$$

$$= 2F_0[-n_t, -M; 1] \hspace{1cm} (31b)$$

5 The PMHT Algorithm: State Covariance Estimates

Analogous to the approach taken by Blanding, Willett, Streit and Dunham [18], who only considered the single-sensor case, we shall show that estimator consistency is improved when the target to measurement assignment probabilities are normalized and the covariance estimate from the MSJPDAF are used. The MSJPDAF is a generalization of the JPDAF to multiple sensors (see, for example [19] for information on the basic JPDAF). There exist two forms of the MSJPDAF, a sequential and a parallel one, which were contrasted by Pao and Frei [20]. Because the sensor fusion is done in parallel at each step of the PMHT, we shall consider the parallel version of the MSJPDAF.

The MSJPDAF requires that the target-measurement association probabilities sum to one over all measurements for a particular target plus the probability that that target was not detected. Noting that in the PMHT measurement model the assignment of one measurement to the target has no bearing on the probability that another measurement is assigned to the same target, the probability that a particular target, \(m_s\), was not detected by a particular sensor, \(s\), at a particular time, \(t\), is given as follows:

$$\beta_{m,0,s}(t) = \prod_{r=1}^{n_t} (1 - w_{m,r}(t, s)) \hspace{1cm} (34)$$

Thus the normalization over the observations, not changing the probability of a missed detection is:

$$\beta_{m,r,s}(t) = w_{m,r}(t, s) \frac{1 - \beta_{m,0,s}(t)}{\sum_{r=1}^{n_t} w_{m,r}(t, s)} \hspace{1cm} (35)$$

The covariance update from the parallel MSJPDAF given by Pao and Frei [20] is:

$$\mathbf{P}_m(t | t) = \sum_C \beta_{m,C}(t) (\mathbf{P}_m,C(t | t) + \mathbf{x}_m,C(t | t)\mathbf{x}_m,C(t | t)^\prime) - \mathbf{x}_m(t | t)\mathbf{x}_m(t | t)^\prime \hspace{1cm} (36a)$$

\(C\) represents a particular combination of assignments between sensors for a particular target. Each \(\beta_{m,C}(t)\) is a product of \(\beta_{m,r,s}(t)\) terms over all sensors for a combination of assignments \(r\), for each sensor. The whole set of \(C\) is the set of all possible measurement to target and clutter assignments at a particular time over all sensors. This means that the covariance calculation is roughly exponentially complex as a function of the number of sensors.

In (36a), \(\mathbf{x}_m,C(t | t)\) represents the state update of the Kalman filter if the measurement-assignment for all sensors given by \(C\) is correct. This means fusing the actual observations in the same way that the synthetic measurements were fused in (20), (21), and (22) and then updating the previous state, \(\mathbf{x}_m(t - 1 | t - 1)\), using these observations. The covariance \(\mathbf{P}_m,C(t | t)\) is likewise what the covariance would be if assignment \(C\) is correct. \(\mathbf{x}_m(t | t)\) is a the weighted average of these other state updates:

$$\mathbf{x}_m(t | t) = \sum_C \beta_{m,C} \mathbf{x}_m,C(t | t) \hspace{1cm} (37)$$

6 Simulation

We compared the track retention rates of the MSPMHT against the MSJPDAF for two targets, whereby we used the sequential version of the MSJPDAF. We used the two dimensional Discrete White Noise Acceleration Model for motion [4]. Two sensors taking Cartesian measurements without classification were used. The ordering of the elements of the state was: \([x, y, \dot{x}, \dot{y}]\). Using a 90% probability of detecting each track at each sensor, the simulation param-

---

3[21] discusses the sequential MSJPDAF algorithm, but provides an incorrect state covariance estimate. [22] provides the correct state covariance estimate when solving a different problem.
eters were as follows:

\[
F = \begin{bmatrix}
1 & 0 & T & 0 \\
0 & 1 & 0 & T \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (38)

\[
Q = \begin{bmatrix}
T^4/4 & 0 & T^3/2 & 0 \\
0 & T^4/4 & 0 & T^3/2 \\
T^3/2 & 0 & T^2 & 0 \\
0 & T^3/2 & 0 & T^2 \\
\end{bmatrix} \sigma_p^2
\] (39)

\[
R = \begin{bmatrix}
1 & 0 & \sigma_m^2 \\
0 & 1 & \sigma_m^2 \\
\end{bmatrix}
\] (40)

The sampling time \( T \) was set to 30 seconds. 500 Monte Carlo runs were performed. The first target was placed at the origin given an initial velocity of \( \frac{7}{2} \) m/s at a 45 degree angle from the \( x \) axis (4.9497 m/s in both the \( x \) and \( y \) coordinates). The second target was assigned the same speed, but having the sign of the \( x \) component flipped. Its initial \( x \) position was 1856m. \( \sigma_m \) was 50m for both sensors and \( \sigma_p \) was 0.02 m². Clutter was generated uniformly in a viewing square having \( x \) and \( y \) bounded between \(-1\)km and \(2.5\)km. The number of clutter points at each step was determined by a Poisson random variable having mean 49. 500 Monte Carlo runs were performed. Initialization was done by giving two correctly assigned observations for each track to an information filter [4]. The simulation lasted 10 steps past the initialization and a growing window with 20 iterations at each step was used for the PMHT. Figure 1a shows typical runs for two targets.

A track was considered lost if at any point the true location of the target was outside of the 99.97 percent confidence interval of the estimated target location. Figure 1b shows the track loss performance. As expected, deterministic annealing significantly improved the track-loss performance of the MSPMHT. Track retention was not significantly affected by the covariance estimation, because the window for the PMHT did not slide.

Notable is the fact that in both multi-sensor scenarios, the use of deterministic annealing significantly improves track retention, agreeing with previous results done using a single sensor on a single track [8] and [9].

7 A Summary of the PMHT

We shall give a summary of the basic PMHT algorithm assuming that all targets have the same probability of detection and that the measurements all have the same measurement matrix, \( H \).

1. Set the initial state estimate for each target at the current time step to the Kalman filter predicted value of the state \( x_m(t|t-1) \).

2. For each sensor and observation, calculate the posterior assignment probabilities, \( w_{k,s}(t), r(t, s) \), according to (12).

3. Create the synthetic measurements, \( \tilde{z}_{m,s}(t) \), with their corresponding measurement covariances, \( \tilde{R}_{m,s}(t) \) for each target, measurement and sensor according to (17) and (18) or (26) and (25).

4. Merge the synthetic measurements between sensors according to (23) and (24).

5. Using the fixed initial state estimate, \( x_m(1) \), and state covariance estimate, \( P_m(1) \) , for each track, run the Kalman smoother using the merged measurements as the observations. Do not smooth the initial state.

6. Update the confusion matrix using (28)

7. Go to step 2. Repeat until convergence of the EM algorithm.

8. If desired, estimate the covariance of the updated state estimate \( x_m(t|t) \) using a consistent approximation, such as (36a).

9. Slide the batch window forward, using the proper covariance estimate for the new initial state, such as in (36a).

10. Go to 1.

8 Conclusion

We derived a general form of the PMHT involving clutter, multiple sensors and classification measurements. We provided a simpler method of performing the maximization step when using multiple sensors and we demonstrated that deterministic annealing can significantly improve tracker performance. Although having worse performance than comparable algorithms, such as the MSPDAF, the PMHT algorithm derived here can form the basis of other modifications of the PMHT, which achieve better track retention.

References


Figure 1: The observations from the first sensor and the last frame of clutter for that sensor are shown in (a) (The axes show distance in meters). Circles are clutter; pluses are from the target. The fraction of tracks not lost at each step are in b.


