

A Critical Look at the PMHT

DAVID F. CROUSE
MARCO GUERRIERO
PETER WILLETT

We combine concepts from numerous papers to provide a derivation and description of a generalized Probabilistic Multi-Hypothesis Tracker that can track multiple targets in a cluttered environment, utilizing multiple sensors and feature measurements, if available. Additionally, we provide a full derivation of the algorithm, including parts omitted or abbreviated in other work. We also provide an improved analytic solution for the prior target-measurement probabilities conditioned on the number of observations, 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, explore the use of deterministic annealing to improve performance, and discuss implementation difficulties.

Manuscript received November 11, 2008; revised April 3, 2009; released for publication April 8, 2009.

Refereeing of this contribution was handled by Yvo Boers.

This research was supported by the Office of Naval Research under contract N00014-07-1-0429, and by Vectrass, Inc.

We would like to thank Yaakov Bar-Shalom for his comments.

Authors' address: Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, U-2157, Storrs, Connecticut 06269, E-mail: ({crouse, marco.guerriero, willett}@engr.uconn.edu).

1557-6418/09/\$17.00 © 2009 JAIF

1. OVERVIEW

Since its creation by Streit and Luginbuhl in 1993 [60], much research has been done on the Probabilistic Multi-Hypothesis Tracker (PMHT). In this paper, we combine concepts from past works and provide a general version of the PMHT algorithm allowing for tracking in the presence of clutter (false alarms) and missed detections and the utilization of classification data, range rate information, and multiple synchronous 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 versions of the PMHT that build upon or alter the basis of the algorithm, such as the Multi-Frame Assignment PMHT (MFPMT) accounting for missed detections by Blanding, Willett, Streit, and Dunham [7]. Being a generalized version of the PMHT, the algorithm might be interchangeably named the PMHT or the Multi-Sensor PMHT (MSPMHT), in line with the naming convention of previous work.

In the subsequent sections, we derive the general form of the PMHT algorithm while discussing implementation difficulties. In Section 2, an overview of previous contributions to the algorithm is provided. Section 3 describes the EM algorithm, which forms the basis of the PMHT. Section 4 derives the state estimates within the PMHT, discussing implementation issues associated with precision problems in 4.3, how and why one might wish to include deterministic annealing to improve performance in 4.3, what to do if each sensor has a different field of view in 4.4, and out-of-order measurement delivery in 4.5. In Section 5, we compare the complexity of the PMHT against that of the Joint Probability Data Association Filter (JPDAF), which is a popular non-batch tracking algorithm.¹ We describe the conditions under which the PMHT has a lower complexity than the MSJPDAF. Section 6 explains how the PMHT, a batch algorithm, can be used over data sequences longer than the batch. Section 7 then discusses state covariance estimation in the PMHT. The algorithm is summarized in Section 8. Section 9 provides a simulation of the PMHT with multiple sensors to verify that the covariance estimation procedure of Section 7 provides, under certain conditions, consistent estimates when multiple targets are used, and demonstrates that deterministic annealing can significantly improve tracker performance when using multiple sensors. Section 10 summarizes the paper. The appendices provide derivations of the target-measurement association probabilities conditioned on the observations, and the target-measurement association probabilities conditioned only on the number of observations.

¹It is also referred to as the Multi-Sensor JPDAF (MSJPDAF) in the multisensor case.

2. PREVIOUS WORK ON THE PMHT ALGORITHM

The Probabilistic Multi-Hypothesis Tracker (PMHT) is a linear-complexity, EM algorithm based, batch target tracking algorithm for use in tracking multiple targets in the presence of clutter when the target-measurement associations are unknown.² Having a memory of the last N scans of data, it attempts to find the maximum a posteriori estimate of the target state in the current scan. This is similar to a later algorithm by Pulford and Logothetis [50], which estimates the target-measurement associations under different measurement models. Being a batch algorithm, it can easily handle delayed measurements, which may simply be added to the batch when they arrive, as was mentioned by Efe, Ruan and Willett [19]. For use as a practical tracker, which must produce track estimates before receiving a full N scans of data, the PMHT can be run at each step on a growing window until a full N scans of data have been acquired, at which point the window slides. This growing and sliding window has been shown to be more effective than other methods by Willett, Ruan, and Streit [70].

The first EM based tracking algorithm was a maximum-likelihood (ML) batch algorithm by Avitzour [2]. This tracker had a high complexity, requiring the calculation of all target-measurement association probabilities. A later algorithm by Molnar and Modestino [41] was a non-batch EM approach to tracking that calculates the maximum a posteriori (MAP) estimate and uses a Markov random field to model the target-measurement associations, resulting in a significantly lower complexity. Jeong and Park [31] used an alternative version of the EM algorithm to produce a recursive MAP target tracker that also estimates various parameters, reducing its complexity by approximating the joint association event probabilities reducing its complexity by approximating multiple target joint association event probabilities as products of single target joint association event probabilities. Pulford and La Scala [49] used the EM algorithm coupled with the Viterbi algorithm to estimate target maneuvers.

In contrast, the PMHT uses an arguably incorrect measurement model in order to reduce its complexity. That is, when told that a particular measurement originated from a particular target, it ignores any conditioning this may imply when determining the posterior association probabilities of the other measurements. As a result, the probability of a particular measurement coming from a particular target is independent of whether that or any other target produced any of the other measurements, and each target is allowed to produce any number of measurements. However, each measurement can only originate from a single target, which is realistic when the targets are well resolved. A version of the PMHT accounting for unresolved targets has also been developed by Davey [11].

It should be noted that the PMHT is not the only algorithm utilizing an “incorrect” measurement model. Particle filter based trackers by Hue, Le Cadre, and Pérez [27], [28] as well as by Gilholm, Godsill, Maskell, and Salmond [25] utilize the same model. The measurement model has some appeal when high resolution sensors are able to over-resolve the target.

The PMHT was first proposed by Streit and Luginbuhl in 1993 [60] with the first full statement of the algorithm appearing in a Naval technical report two years later [61]. The PMHT algorithm was defined very generally in [61], allowing for the target dynamics and measurement model to have arbitrary distributions. However, such a generic model did not allow for the maximization step of the EM algorithm, upon which it is based, to be easily performed, and thus the practical implementation presented had a discrete-time linear motion model, as used in the Kalman filter. Since then, quite a few variants of the PMHT algorithm have emerged, many of which have been compared by Willett, Ruan, and Streit in [71].

The best performing variants of the PMHT, that is those having better track-loss characteristics than the JPDAF, are the Turbo PMHT, by Ruan and Willett [56] and Willett, Ruan, and Streit [69] and Multi-Frame PMHT (MFPMHT) algorithms. The currently best-performing version of the PMHT is the MFPMHT that accounts for missed detections, by Blanding, Willett, Streit, and Dunham [7], which is a modification of an earlier MFPMHT by Streit [59]. However, the better performance of the MFPMHTs comes with an increased complexity, being roughly exponentially complex over the last L frames of an N frame batch.

The homeothetic PMHT, first derived by Rago, Willett, and Streit in [53], was an *ad-hoc* approach to improving the performance of the PMHT through the use of multiple measurement models with different noise covariances. The different covariances were intended to overcome estimation problems in the PMHT and not to function as multiple models for states inherent to the targets. However, this may be thought of as a forerunner to multiple model PMHT algorithms. Logothetis, Krishnamurthy, and Holst [35]³ were the first to develop a form of the PMHT algorithm involving multiple models to account for target maneuvers. The transition between model states was governed by a Hidden Markov Model (HMM). The maneuvers were handled using additional unknown “control” inputs to the Kalman filter. Willett, Ruan, and Streit [69] also did this, modeling the maneuvers as increases in the process noise of the targets forming the MPMHT. In both approaches, the active model of the target was modeled as one of the “nuisance” variables in the EM algorithm. Pulford and La Scala [49] took a different approach, making the maneuvers part of the quantity to be estimated in the EM algorithm. Their maneuver-estimation approach can be used

²A though the focus is on target tracking, the PMHT algorithm has found use in other applications, such as cartography [10].

³This is the journal version of an earlier conference paper [34].

with various EM-algorithm based trackers including the PMHT. Willett, Ruan, and Streit created an interactive multiple model approach to tracking using the PMHT in the presence of maneuvers (IMM-PMHT), replacing the backward-forward algorithm used in [69] with an interacting multiple model (IMM). They also derived a turbo-coding based extension to the MPMHT, which they dubbed the Turbo MPMHT. This later matured into the Turbo PMHT, as described by Ruan and Willett [56]. More recently, Luginbuhl, Ainsleigh, Mathews, and Streit [36] demonstrated how to derive the observed data likelihood function for the family of manoeuvring PMHT trackers.

The basic PMHT is an algorithm that tracks targets based upon discrete observations at each scan. However, a variant called the Histogram PMHT (H-PMHT) allows the PMHT to process continuous data directly from a sensor. The concept was first introduced by Luginbuhl and Willett [37] as a method of tracking a general frequency-modulated signal in noise (explained in more detail in [38]), and a variant appeared in [58]. Walsh, Graham, Streit, Luginbuhl, and Mathews [64] later presented a one-dimensional application of the algorithm. Pakfiliz and Efe [44] presented a two-dimensional application, and most recently, Davey, Rutten, and Cheung [15] compared it against other track-before-detect methods. In this paper, we do not consider the processing of continuous sensor measurements. The use of the PMHT in tracking problems with bearing-only measurements has been studied by Giannopoulos, Streit, and Swaszek [23].

In its original version, the PMHT did not account for clutter. Rago, Willett, and Streit [52] extended the PMHT to cluttered environments under a number of assumptions, regarding the probability of a measurement originating from clutter, by modifying the target measurement assignment probabilities $w_{k,(t),l}(t)$. In the next year, Hutchins and Dunham produced a similar version of the PMHT for use in cluttered environments [29], involving an *ad-hoc* constant in the denominator of the target measurement assignment probabilities. In later publications, an analytically derived solution has been used, but no complete derivation has been given. In this paper, we provide an explicit derivation of the PMHT including clutter, and we provide a full Bayesian derivation of the prior and posterior association probabilities $\pi_{k,(t)}(n_t)$ and $w_{k,(t),l}(t)$. The probabilities $\pi_{k,(t)}(n_t)$ have previously been derived by Wieneke and Koch [66], but here it is developed in such a way that the solution could be simplified by omitting “fictitious targets” that had been used in [66].

Davey, Gray, and Streit [14] introduced the use of target classification measurements into the PMHT. Namely, extra data can be used to identify the type of each observations, e.g., whether an observation is clutter, a plane or a missile. A more complete analysis of this work is given in Davey’s PhD thesis [9]. In this paper, we show how classification measurements can be used in a multisensor environment.

The simplest approach to multisensor tracking with the PMHT was first considered by Rago, Willett, and Streit [51]. They pooled all of the measurements from all of the sensors together and ran the PMHT as if all of the measurements came from a single sensor. One can justify this by the fact that the PMHT’s measurement model allows for a single target to produce multiple observations. Hempel [26] considered the robustness of the PMHT to registration errors when the measurements from all sensors are pooled. However, versions of the PMHT specifically accounting for multiple sensors by modifying the likelihood function to reflect their presence have been developed, and have been shown to improve the performance of the tracker over the pooled measurement approach. These were developed concurrently by Krieg and Gray [33], by Giannopoulos, Streit, and Swaszek [24] and by Gauvrit, Le Cadre, and Jaufret [22]. All of these derivations used the Levenberg-Marquadt method (described, for example in [5]) for performing the maximization step of the EM algorithm. In this paper, we show that a simpler, non-iterative approach exists.

In its original form, the PMHT algorithm was meant to track a known number of targets and lacked any notion of track discovery, termination or merging. However, such tasks are necessary for a tracker to be usable in real-world situations. Several advances have been made in integrating track discovery and termination features into the PMHT. A complete track management system, in which tracks were discovered and terminated by separate algorithms outside of the PMHT algorithm, was first introduced by Luginbuhl, Sun, and Willett [39], whereby track extraction was done via the Hough transform. Alexiev [1] also considered the use of the Hough transform with the PMHT. Davey and Gray [12] later gave a comparison of various methods of track initiation, and Dunham and Hutchins [17] considered using the MHT as a track-finding front-end to the PMHT, whereby once tracks were stable they were handed off to the PMHT. Since then, however, additional methods of track management have emerged. Davey and Gray introduced the Hysteresis PMHT [13], which treats the existence of a target as an extra state in the estimator. Mušicki and Wang [43] used an *ad-hoc* approach of modifying the posterior association probabilities to do the same thing. Wieneke and Willett [68] looked at methods of determining track deletion and Wieneke and Koch looked at hypothesis tests for estimating the number of tracks present [67].

3. THE EXPECTATION MAXIMIZATION ALGORITHM AND DETERMINISTIC ANNEALING

The PMHT is based on the EM algorithm. The EM algorithm, discovered by Dempster, Laird, and Rubin [16], is a method of determining the ML or MAP estimate of data given incomplete information. Redner and Walker [54] specifically looked at the use of the

EM algorithm for ML estimation of parameters of mixture densities, a topic that is relevant to EM algorithm based target tracking. The EM algorithm is summarized here, which is extensively covered in the monograph by McLachlan and Krishnen [40], and we must note the tutorial by Moon in [42]. It should be noted that the EM algorithm does not provide the covariance of its estimate. This is, however, necessary for the tracker to be useful and is discussed in Section 7.

Let \mathbf{X} be an unknown random quantity the MAP estimate of which 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

$$\hat{\mathbf{X}}_{\text{MAP}} = \arg \max_{\mathbf{X}} E\{\log(p(\mathbf{X} | \mathbf{Z}))\}, \quad (1)$$

in which p 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:

$$Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) \triangleq \int_{\mathbf{K}} \log(p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})) p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z}) d\mathbf{K}. \quad (2)$$

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

$$\mathbf{X}^{(n+1)} = \arg \max_{\mathbf{X}^{(n+1)}} Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}). \quad (3)$$

n is then incremented and one continues until a desired level of convergence has been attained.

In some instances, the PDF $p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})$ may be difficult to determine. If this is the case, by the definition of conditional expectation, it can be noted that

$$p(\mathbf{X}, \mathbf{K} | \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{X}, \mathbf{K})}{p(\mathbf{Z})}. \quad (4)$$

Substituting (4) into (2) and separating the logarithm we get

$$Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) = \int_{\mathbf{K}} \log(p(\mathbf{Z}, \mathbf{X}^{(n+1)}, \mathbf{K})) p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z}) d\mathbf{K} - \log p(\mathbf{Z}). \quad (5)$$

Because $p(\mathbf{Z})$ is a constant, that term may be dropped from (5), since it has no effect on the location of the maximum and thus $p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})$ and $p(\mathbf{Z}, \mathbf{X}^{(n+1)}, \mathbf{K})$ may be used interchangeably in the EM algorithm.

Boyles [8] and Wu [30] studied the convergence properties of the EM algorithm, correcting a mistake

in Theorem 2 of Dempster, Laird, and Rubin's original paper [16]. They showed that the EM algorithm is guaranteed to converge to a saddle point or a local maximum, which need not be the desired global maximum. To which critical point it converges is highly dependent upon the initial estimate $\mathbf{X}^{(1)}$.

Over the years, a number of versions of the EM algorithm have been developed, many of which are summarized by McLachlan and Krishnen [40] and by Roche [55]. Most sought to increase the convergence speed of the algorithm. However, for the PMHT algorithm, the primary concern is avoiding convergence to local maxima. In order to reduce dependence on the initial estimate and encourage convergence to the global maximum, the Deterministic Annealing (DA) EM algorithm was developed by Ueda and Nakano [62], who recognized that solving the maximum likelihood problem is analogous to similar problems linking concepts in thermodynamics and information theory. This was applied to the PMHT first in 1999 by Strandlie and Zerubia [57] and was later applied in a more general form by Wieneke and Koch [66]. When tracking a single target, the basic PMHT algorithm with deterministic annealing is identical to the Deterministic Annealing Filter by Frühwirth and Strandlie [20]. As shall be described after the basic derivation of the PMHT, deterministic annealing can be added to almost any version of the PMHT algorithm. To derive the DA-EM algorithm, we shall use the definition of conditional expectation to note that

$$p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^{(n)})}{\int_{\mathbf{K}_1} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^{(n)}) d\mathbf{K}_1}. \quad (6)$$

The denominator in (6) is equal to $p(\mathbf{Z}, \mathbf{X})$. The DA-EM algorithm substitutes (6) into (2) and introduces the term β as

$$Q_{\text{DA}}(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) \triangleq \int_{\mathbf{K}} \log(p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})) \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^{(n)})^\beta}{\int_{\mathbf{K}_1} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^{(n)})^\beta d\mathbf{K}_1} d\mathbf{K}. \quad (7)$$

In the original description of the DA-EM algorithm, β , the inverse of which corresponds to the "temperature" in an analogous thermodynamic problem, is initially set to a value between 0 and 1. Q_{DA} is then iterated with respect to $\mathbf{X}^{(n+1)}$ and $\mathbf{X}^{(n)}$ until convergence, as is done in the regular EM algorithm. Then β is increased to a value closer to one and Q_{DA} is again iterated until convergence. The DA-EM algorithm is complete when β has finally been increased to 1. Note however, that the original version of the DA-EM algorithm did not specify exactly how β was to be increased.

When $\beta < 1$, the PDF $p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z})$ becomes flatter, which reduces dependence of the algorithm on $\mathbf{X}^{(n)}$. The reasoning behind the DA-EM algorithm is that by slowly increasing β , the effect of $p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z})$ is increased at the same time that the estimate $\mathbf{X}^{(n)}$ improves.

In the final step $\beta = 1$ and (7) is equivalent to (2) and the EM algorithm should be more likely to converge to the global MAP estimate, because of the improved prior estimate $\mathbf{X}^{(n)}$.

In order to use the DA-EM algorithm in a practical implementation, one must have a method of increasing β . If β is increased very slowly, then in general, one iteration should be enough for convergence at each value of β . Thus, in [57] and [66], the DA-EM algorithm was carried out as follows:

Let n_{\max} be the number of iterations that one wishes to do. For each iteration from $n = 1$ onwards, set $\beta = n/n_{\max}$. Now iterate the EM algorithm as would normally be done. That is, \mathbf{X}^{n+1} is the set to the value maximizing $Q_{\text{DA}}(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)})$ at each step. In the final iteration $\beta = 1$ and the result is the EM algorithm result.

Although convergence to the global MAP estimate is not guaranteed, as long as n_{\max} is large enough, this approach will generally outperform the basic EM algorithm. In Section 9, we demonstrate how deterministic annealing improves tracking performance when multiple sensors are used. Although the above method is the procedure needed to get the MAP estimate, the ML estimate can be attained by replacing $p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})$ with $p(\mathbf{Z}, \mathbf{K} | \mathbf{X}^{(n+1)})$.

4. THE PMHT ALGORITHM: STATE ESTIMATES

We shall now derive a general form of the PMHT algorithm allowing for the presence of clutter, multiple synchronous sensors and the use of classification information. The most general form of the PMHT allows for a very generic target motion and measurement model [61]. However, due to the difficulty of the maximization step of the EM algorithm under a generic model, practical implementations of the PMHT are often based upon the motion models that, in the absence of target-measurement association uncertainty, contain the assumptions inherent to the basic Kalman filter (see, e.g., [4]). We shall derive the PMHT under such a model, accounting for clutter, taking advantage of multiple sensors, and utilizing classification data. It shall be assumed that the measurement noise between sensors is uncorrelated and that sensors have the same field of view. The case where the sensors have different fields of view and gating is present, is discussed in the following section.

Given M targets, the state vector at time t for the m th target shall be designated as \mathbf{x}_m . The observation originating from the m th target shall be designated $\mathbf{y}_m(t)$. The basic discrete-time kinematic motion and observations equations are given by⁴

$$\mathbf{x}_m(t+1) = \mathbf{F}_m(t)\mathbf{x}_m(t) + \mathbf{v}_m(t) \quad (8)$$

⁴The Kalman filter and Kalman smoother equations associated with this model are summarized in Appendix D.

and

$$\mathbf{y}_m(t) = \mathbf{H}_m(t)\mathbf{x}_m(t) + \mathbf{w}_m(t). \quad (9)$$

The process noise at time t , $\mathbf{v}_m(t)$, is assumed to be Gaussian distributed with zero mean and covariance $\mathbf{Q}_m(t)$. The measurement noise $\mathbf{w}_m(t)$ is also modeled as a zero-mean Gaussian random variable with covariance $\mathbf{R}_m(t)$ and is assumed to be uncorrelated with the process noise. The covariance of the true measurement from the m th target $\mathbf{R}_m(t)$ describing $\mathbf{w}_m(t)$ from (9), corresponds to the covariance of one of the measurements out of the set of all measurements at time t , whereby $\mathbf{R}_{r,s}(t)$ shall represent the covariance of measurement r from sensor s based upon the location of the observation, without stating a particular associated target.

Let \mathbf{Z} be all of the measurements and classification information from time $t = 1$ to N . Let \mathbf{X} be the states of all of the targets over the same time period and \mathbf{K} be the set of associations between targets and measurements. Let there be a total of S sensors that take measurements synchronously. If $\mathbf{z}_{r,s}(t)$ is the measurement r at time t from sensor s that 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 \mathbf{X} without explicitly determining which set of $k_{r,s}(t)$ from the set of all possible target to measurement associations, \mathbf{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 [14]. We shall assume that some type of classification has already been done for each measurement, giving us $z_{r,s}^C(t)$, 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 \mathbf{C} whose elements are defined as

$$c(i, m) = \Pr(z_{r,s}^C(t) = i | k_{r,s}(t) = m). \quad (10)$$

i in (10) 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, which is why $c_{i,m}$ is not indexed against time.⁵ That is, the appearance of each target is assumed to be constant. It shall also be assumed independent of that state. 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 \mathbf{X} in the PMHT algorithm. Thus, $\arg \max_i c_{i,m}$ will be the MAP estimate of the classification of target m at the end of the algorithm.

Let us find the first PDF in (5), including \mathbf{C} next to \mathbf{X} as an unknown to be estimated. Let $n_t(s)$ be the number of measurements at time t that came from

⁵This assumption plays a role in our subsequent estimation of the confusion matrix via the EM algorithm as part of the PMHT. However, if one does not wish to perform EM algorithmic estimation of the confusion matrix, then a time-varying confusion matrix may be used without modification to the rest of the PMHT.

sensor s . In order for $p(\mathbf{Z}, \mathbf{X}, \mathbf{K}, \mathbf{C})$ to be written, it shall be conditioned on $n_t(s)$. However, we will not explicitly write this conditioning except when necessary. $p(\mathbf{Z}, \mathbf{X}, \mathbf{K}, \mathbf{C})$ is given as

$$p(\mathbf{Z}, \mathbf{X}, \mathbf{K}, \mathbf{C}) = p(\mathbf{X})p(\mathbf{Z}, \mathbf{K}, \mathbf{C} | \mathbf{X}) \quad (11a)$$

$$\begin{aligned} &= \overbrace{\left(\prod_{m=1}^M p(\mathbf{x}_m(1)) \prod_{t_x=2}^N p(\mathbf{x}_m(t_x) | \mathbf{x}_m(t_x - 1)) \right)}^{p(\mathbf{X})} \\ &\times \overbrace{\prod_{s=1}^S \prod_{t=1}^N \prod_{r=1}^{n_t(s)} \Pr(k_{r,s}(t) | \mathbf{x}_{k_{r,s}(t)}(t), n_t) p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t))}^{p(\mathbf{Z}, \mathbf{K} | \mathbf{X})} \\ &\times \underbrace{c(\mathbf{z}_{r,s}^C(t), k_{r,s}(t))}_{p(\mathbf{C} | \mathbf{Z}, \mathbf{X}, \mathbf{K})}. \quad (11b) \end{aligned}$$

In (11b), the PDF $p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t))$ depends upon whether the measurement came from clutter or from a target and is given by

$$p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t)) = \begin{cases} \mu(t, \mathbf{z}_{r,s}(t)) & \text{if } k_{r,s}(t) = 0 \\ \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\} & \text{if } k_{r,s}(t) \neq 0 \end{cases}. \quad (12)$$

In (12), μ denoted the PDF of the clutter, which we shall assume to be continuous as a function of $\mathbf{z}_{r,s}(t)$, and which need not be uniform, and $\hat{\mathbf{y}}_{k_{r,s}(t)}(t)$ is the estimate of \mathbf{y} from (9). That is,

$$\hat{\mathbf{y}}_{k_{r,s}(t)}(t) = \mathbf{H}_{k_{r,s}(t)}(t) \mathbf{x}_{k_{r,s}(t)}(t). \quad (13)$$

Define $w_{k_{r,s}(t),r}(t,s) \triangleq \Pr(k_{r,s}(t) | \mathbf{x}_{k_{r,s}(t)}(t), \mathbf{Z}(t), \mathbf{C}, n_t(s))$ as the probability of a particular measurement-target assignment at time t , whereby clutter is target $k_{r,s}(t) = 0$. We shall refer to these as the ‘‘posterior association probabilities.’’ One instance of \mathbf{K} defines $k_{r,s}(t)$ over all

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 (5), may be obtained directly by multiplying the marginal probabilities over all time and measurements for all of the assignments:

$$p(\mathbf{K} | \mathbf{X}, \mathbf{C}, \mathbf{Z}) = \prod_{s=1}^S \prod_{t=1}^N \prod_{r=1}^{n_t(s)} w_{k_{r,s}(t),r}(t,s). \quad (15)$$

In order to make the notation in the above equation correct if there are no observations at a particular sensor at a certain time, that is if $n_t(s) = 0$, the following definition must be used:

$$\prod_{r=1}^0 w_{k_{r,s}(t),r}(t,s) \triangleq 1. \quad (16)$$

We would now like to determine the posterior association probability $w_{k_{r,s}(t),r}(t,s)$. In previous published works on the PMHT, no formal derivation of this in the presence of clutter has been done and because it is not immediately obvious, it shall be included here for completeness, in Appendix A. Define $\pi_{k_{r,s}(t)}(n_t(s), t) \triangleq \Pr(k_{r,s}(t) = m | \mathbf{x}_m(t), n_t(s))$. This shall be referred to as a ‘‘prior association probability.’’⁶ The $\pi_{k_{r,s}(t)}(n_t(s), t)$ values were derived using ‘‘imaginary’’ targets by Wieneke and Koch [66] and we have re-derived them in a simpler form in Appendix B. The original PMHT algorithm made them a parameter to be estimated by the EM algorithm.

Using the solution from Appendix A for $k_{r,s}(t) \neq 0$, that is when measurement r from sensor s is not clutter, then the posterior association probabilities for non-clutter targets are as follows (in the final solution to the PMHT algorithm, it will turn out that one never needs to evaluate $w_{0,r}(t,s)$):

$$w_{k_{r,s}(t),r}(t,s) = \frac{\pi_{k_{r,s}(t)}(n_t(s), t) \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\} c(\mathbf{z}_{r,s}^C(t), k_{r,s}(t))}{\pi_0(n_t(s), t) \mu(t, \mathbf{z}_{r,s}(t)) c(\mathbf{z}_{r,s}^C(t), 0) + \sum_{m=1}^M \pi_m(n_t(s), t) \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\} c(\mathbf{z}_{r,s}^C(t), m)}. \quad (17)$$

measurements r and sensors s , for all time in the batch. The sum over \mathbf{K} is equal to the sum over all sensors of the sum over \mathbf{K}_s , which is defined as

$$\sum_{\mathbf{K}_s} (\cdot) = \sum_{k_{1,s}(1)=0}^M \sum_{k_{2,s}(1)=0}^M \cdots \sum_{k_{n_1(s),s}(1)=0}^M \sum_{k_{1,s}(2)=0}^M \cdots \sum_{k_{n_N(s),s}(N)=0}^M (\cdot). \quad (14)$$

In a simple model where all targets have the same probability of detection, $P_D(s)$, when viewed by sensor s , and the number of clutter points at each sensor

⁶‘‘prior association probability’’ is somewhat of a misnomer, since $\pi_{k_{r,s}(t)}(n_t(s), t)$ is conditioned on the number of observations. However, this naming convention helps differentiate it from the posterior association probabilities.

is Poisson distributed with mean $\lambda(s)V(s)$ where $\lambda(s)$ represents the mean amount of clutter per unit volume at sensor s and $V(s)$ is the volume of the viewing area for that sensor, then, as derived in the Appendices B and C, the prior association probabilities are the same for all non-clutter targets and may be divided across the numerator and denominator giving us the following expression for $w_{k_{r,s}(t),r}(t,s)$

$$w_{k_{r,s}(t),r}(t,s) = \frac{\mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\} c(z_{r,s}^C(t), k_{r,s}(t))}{\bar{\pi}_s(n_t(s), t) \mu(t, \mathbf{z}_{r,s}(t)) c(z_{r,s}^C(t), 0) + \sum_{m=1}^M \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\} c(z_{r,s}^C(t), m)}, \quad (18)$$

$$\bar{\pi}_s(n_t(s), t) = -M - \frac{{}_2F_0 \left[-M, -n_t(s); \frac{P_D(s)}{(1-P_D(s))\lambda(s)V(s)} \right]}{{}_2F_0 \left[1-M, 1-n_t(s); \frac{P_D(s)}{(1-P_D(s))\lambda(s)V(s)} \right]}, \quad (19)$$

here the function ${}_2F_0[a_1, a_2; z]$ is a hypergeometric function.

Combining (15) and (11b) and omitting the constant $p(\mathbf{Z})$ we may form the Q function for the basic EM algorithm in (5):

Equation (21) comes from the fact that at any given time the observations must have originated from a target or from clutter. $\mathbf{K}_s \setminus k_{r_1,s}(t_i)$ represents the set of all assignments involving sensor s except for $k_{r_1,s}(t_i)$ and (22) comes directly from the definition of $w_{k_{r,s}(t),r}^n(t,s)$. Note that when $k_{r,s}(t) = 0$, that is when observation r is clutter, $p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t))$ contains no terms

involving \mathbf{X} . Therefore, for purposes of maximizing Q , we may omit all clutter terms from the second set of sums, because they disappear when the derivative is taken. Using this fact and (21) and (22), Equation (20b) may be simplified as follows:

$$Q(\mathbf{X}^{n+1}, \mathbf{C}^{n+1}; \mathbf{X}^n, \mathbf{C}^n) = \sum_{\mathbf{K}} \log(p(\mathbf{Z}, \mathbf{X}^{(n+1)}, \mathbf{C}^{(n+1)}, \mathbf{K})) p(\mathbf{K} | \mathbf{X}^n, \mathbf{C}^n, \mathbf{Z}) \quad (20a)$$

$$\begin{aligned} &= \log \left(\prod_{m=1}^M p(\mathbf{x}_m^{(n+1)}(1)) \prod_{t_x=2}^N p(\mathbf{x}_m^{(n+1)}(t_x) | \mathbf{x}_m^{(n+1)}(t_x - 1)) \right) \\ &+ \sum_{\mathbf{K}} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \log(\pi_{k_{r,s}(t)}(n_t(s), t)) \prod_{s_1=1}^S \prod_{t_1=0}^N \prod_{r_1=1}^{n_{t_1}(s_1)} w_{k_{r_1,s_1}(t_1),r_1}^{(n)}(t_1, s_1) \\ &+ \sum_{\mathbf{K}} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \log(c(z_{r,s}^C(t), k_{r,s}(t))) \prod_{s_1=1}^S \prod_{t_1=0}^N \prod_{r_1=1}^{n_{t_1}(s_1)} w_{k_{r_1,s_1}(t_1),r_1}^{(n)}(t_1, s_1) \\ &+ \sum_{\mathbf{K}} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \log(p(\mathbf{z}_{r,s}(t) | k_r(t), \mathbf{x}_{k_{r,s}(t)}^{(n+1)}(t))) \prod_{s_1=1}^S \prod_{t_1=0}^N \prod_{r_1=1}^{n_{t_1}(s_1)} w_{k_{r_1,s_1}(t_1),r_1}^{(n)}(t_1, s_1). \quad (20b) \end{aligned}$$

The superscripts in parentheses in (20b) indicate whether the values in question are to be calculated using the current or the previous estimate of \mathbf{X} in the EM algorithm. As pointed out by Davey [9] in his thesis, (20b) is simplified by use of the two identities

$$\sum_{\mathbf{K}_s} \prod_{t=1}^N \prod_{r=1}^{n_t} w_{k_{r,s}(t),r}(t,s) = 1 \quad (21)$$

and

$$\sum_{\mathbf{K}_s \setminus k_{r_1,s}(t_i)} \prod_{t=1}^N \prod_{r=1}^{n_t} w_{k_{r,s}(t),r}(t,s) = w_{k_{r_1,s}(t_i),r_1}(t_i, s). \quad (22)$$

$$\begin{aligned} Q(\mathbf{X}^{n+1}; \mathbf{X}^n) &= \left. \begin{aligned} &\sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \sum_{m=1}^M \log(\pi_m(n_t(s), t)) w_{m,r}^{(n)}(t, s) \Big\} Q_{\Pi} \\ &+ \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \sum_{m=1}^M \log(c(z_{r,s}^C(t), m)) w_{m,r}^{(n)}(t, s) \Big\} Q_{\mathcal{C}} \end{aligned} \right\} Q_{\mathbf{X}} \\ &+ \left. \begin{aligned} &\log \left(\prod_{m=1}^M p(\mathbf{x}_m^{(n+1)}(1)) \prod_{t_x=2}^N p(\mathbf{x}_m^{(n+1)}(t_x) | \mathbf{x}_m^{(n+1)}(t_x - 1)) \right) \\ &+ \sum_{s=1}^S \sum_{t=1}^N \sum_{m=1}^M \sum_{r=1}^{n_t(s)} \\ &\times \log(\mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m^{(n+1)}(t), \mathbf{R}_{r,s}(t)\}) w_{m,r}^{(n)}(t, s) \end{aligned} \right\} Q_{\mathbf{X}}. \quad (23) \end{aligned}$$

The maximization of the state component of the Q -function, $Q_{\mathbf{x}}$, from (23), is performed indirectly by finding an equation with the same gradient, $\nabla_{\mathbf{x}^{n+1}} Q_{\mathbf{x}}$, and thus the same inflection points. Omitting constant terms, the derivative taken over the innermost sum may be transformed as follows:

sensors but no data observation uncertainty. The second term of (27) may be rewritten as follows:

$$-\frac{1}{2} \sum_{t=1}^N \sum_{m=1}^M (\tilde{\mathbf{z}}_m - \mathbf{H}_m \mathbf{x}_m^{(n+1)}(t))' \tilde{\mathbf{R}}_m(t)^{-1} (\tilde{\mathbf{z}}_m - \mathbf{H}_m \mathbf{x}_m^{(n+1)}(t)). \quad (28)$$

$$\nabla_{\mathbf{x}} \left[\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) (\mathbf{z}_{r,s}(t) - \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t)) \mathbf{R}_{r,s}(t)^{-1} (\mathbf{z}_{r,s}(t) - \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t))' \right] \quad (24a)$$

$$= \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} (\mathbf{z}_{r,s}(t) - \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t)) \quad (24b)$$

$$= \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) - \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t) \quad (24c)$$

$$= \left(\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \right) \times \left(\left(\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \right) - \mathbf{x}^{(n+1)}(t) \right) \quad (24d)$$

$$= \tilde{\mathbf{R}}_{m,s} (\tilde{\mathbf{z}}_{m,s} - \mathbf{x}^{(n+1)}(t)) \quad (24e)$$

$$= \nabla_{\mathbf{x}} [(\tilde{\mathbf{z}}_{m,s}(t) - \mathbf{x}^{(n+1)}(t)) \tilde{\mathbf{R}}_{m,s}(t)^{-1} (\tilde{\mathbf{z}}_{m,s}(t) - \mathbf{x}^{(n+1)}(t))']. \quad (24f)$$

The synthetic measurements $\tilde{\mathbf{z}}_{m,s}(t)$ and $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ are defined by

$$\tilde{\mathbf{R}}_{m,s}(t)^{-1} = \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \quad (25)$$

and

$$\tilde{\mathbf{z}}_{m,s}(t) = \tilde{\mathbf{R}}_{m,s}(t) \left(\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}_{r,s}' \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \right). \quad (26)$$

Note that $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ may not be invertible, so a pseudoinverse may be necessary. The equivalency between (24a) and (24f) means that $Q_{\mathbf{x}}$ from (23) has the same derivative as

$$\begin{aligned} & \hat{Q}(\mathbf{X}^{n+1}; \mathbf{X}^n) \\ &= \log \left(\prod_{m=1}^M p(\mathbf{x}_m^{(n+1)}(1)) \prod_{t=2}^N p(\mathbf{x}_m^{(n+1)}(t) | \mathbf{x}_m^{(n+1)}(t-1)) \right) \\ & - \frac{1}{2} \sum_{s=1}^S \sum_{t=1}^N \sum_{m=1}^M (\tilde{\mathbf{z}}_{m,s} - \mathbf{x}_m^{(n+1)}(t))' \\ & \times \tilde{\mathbf{R}}_{m,s}(t)^{-1} (\tilde{\mathbf{z}}_{m,s} - \mathbf{x}_m^{(n+1)}(t)). \end{aligned} \quad (27)$$

Equation (27) is the joint likelihood function of M targets for which there are observations from multiple

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

$$\tilde{\mathbf{z}}_m(t) = [\tilde{\mathbf{z}}_{m,1}(t), \tilde{\mathbf{z}}_{m,2}(t), \dots, \tilde{\mathbf{z}}_{m,S}(t)], \quad (29)$$

$$\mathbf{H}_m = [\mathbf{I}_{x_m,1}, \mathbf{I}_{x_m,2}, \dots, \mathbf{I}_{x_m,S}], \quad (30)$$

and

$$\tilde{\mathbf{R}}_m(t) = \text{diag}[\tilde{\mathbf{R}}_{m,1}, \tilde{\mathbf{R}}_{m,2}, \dots, \tilde{\mathbf{R}}_{m,S}]. \quad (31)$$

Substituting equation (28) into (27) is equivalent to a single-sensor system with no data association uncertainty having measurements given by (29), (30), and (31). 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 (the equations for the Kalman smoother are summarized in Appendix D). This is a simpler approach than using the Levenberg-Marquardt nonlinear regression procedure, as suggested by Giannopoulos, Streit, and Swaszek [24] in the original derivation of the PMHT with multiple sensors. The use of the Kalman smoother was also present in the original single sensor PMHT algorithm.

This method of stacking measurements is a common method of measurement fusion for the Kalman filter when there is no target-measurement association uncertainty. Gan and Harris [21] showed that if at a

particular time for a particular track all sensors have the same measurement matrix, which being $\mathbf{I}_{\mathbf{x}_{m,1}}$ 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 by

$$\tilde{\mathbf{z}}_m(t) = \left(\sum_{s=1}^S \tilde{\mathbf{R}}_{m,s}(t)^{-1} \right)^{-1} \sum_{s=1}^S \tilde{\mathbf{R}}_{m,s}(t)^{-1} \tilde{\mathbf{z}}_{m,s}(t) \quad (32)$$

and

$$\tilde{\mathbf{R}}_m(t) = \left(\sum_{s=1}^S \tilde{\mathbf{R}}_{m,s}(t)^{-1} \right)^{-1}. \quad (33)$$

The Kalman smoother may be thought of as running a forward Kalman filter, and then running a backwards smoothing operation on the track estimate resulting from the Kalman filter step. Note that the use of the pseudoinverse in (26) may be completely avoided if the information filter (described, for example, in [4]) is used in place of the Kalman filter in the first half of the Kalman smoother. The information filter calls for $\tilde{\mathbf{R}}_{m,s}(t)^{-1} \tilde{\mathbf{z}}_{m,s}(t)$, obviating the need to invert $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ in (26).

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, the steps leading up to (24f) may be simplified, resulting in the following simplified synthetic measurements

$$\tilde{\mathbf{z}}_{m,s}(t) = \left(\sum_{r=1}^{n_r(s)} w_{m,r}^{(n)}(t,s) \mathbf{R}_{r,s}(t)^{-1} \right)^{-1} \sum_{r=1}^{n_r(s)} w_{m,r}^{(n)}(t,s) \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \quad (34)$$

and

$$\tilde{\mathbf{R}}_{m,s}(t) = \left(\sum_{r=1}^{n_r(s)} w_{m,r}^{(n)}(t,s) \mathbf{R}_{r,s}(t)^{-1} \right)^{-1}. \quad (35)$$

The form of the synthetic measurements in (34) and (35), allowing for each measurement to have a different covariance matrix was first given in [66]. Previous versions assumed that all measurements have the same covariance. The forms given in (25) and (26) allowing for different measurement matrices, as occurs with doppler measurements, are unique to this paper. Note that if each sensor has a different measurement matrix, then the measurement fusion method given in (32) and (33) is no longer optimal. In this case, the merged measurement given in (29) and (31) should be used with the modified merged measurement matrix,

$$\mathbf{H}_m = [\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_S] \quad (36)$$

where \mathbf{H}_s is the measurement matrix of the s th sensor.

The maximization of the confusion matrix via the gradient $\nabla_{\mathbf{c}_{n+1}} \mathcal{Q}_C$ from (23) is performed under the constraint

$$\sum_{i=1}^{M_C} c_{i,m} = 1. \quad (37)$$

Using (23) and (37), the Lagrangian to maximize is:

$$L_C = \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r(s)} \sum_{m=1}^M \log(c(z_{r,s}^C(t), m)) w_{m,r}^{(n)}(t,s) + \sum_{m=1}^M \lambda_m^C \left(1 - \sum_{i=1}^{M_C} c(i, m) \right) \quad (38a)$$

$$= \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r(s)} \sum_{m=1}^M \sum_{i=1}^{M_C} \delta(z_{r,s}^C - i) \log(c(i, m)) w_{m,r}^{(n)}(t,s) + \sum_{m=1}^M \lambda_m^C \left(1 - \sum_{i=1}^{M_C} c(i, m) \right). \quad (38b)$$

Equation (38a) is equivalent to (38b), where $\delta(t)$ is the Kronecker Delta function, which is one for $t = 0$ and zero otherwise. Differentiating (38b) with respect to a particular $c(i, m)$ gives

$$c_{i,m} = \frac{1}{\lambda_m^C} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} \delta(z_{r,s}^C - i) w_{m,r}^{(n)}(t,s). \quad (39)$$

Applying the constraint given in (37) gives

$$\lambda_m^C = \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} \sum_{i=1}^{M_C} \delta(z_{r,s}^C - i) w_{m,r}^{(n)}(t,s) \quad (40a)$$

$$= \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} w_{m,r}^{(n)}(t,s). \quad (40b)$$

Combining (39) and (40b) gives us the update for the $c_{i,m}$:

$$c_{i,m} = \frac{\sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} \delta(z_{r,s}^C - i) w_{m,r}^{(n)}(t,s)}{\sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} w_{m,r}^{(n)}(t,s)}. \quad (41)$$

4.1. Regarding the Kalman Smoothing Step

The equations for the Kalman smoother are given in Appendix D. It should be noted that although the EM algorithm might call for the initial state estimate for each track, $\mathbf{x}_m(1)$, to be smoothed along with the rest, practically the algorithm is not usable in this manner. The first part of the Kalman smoother is done by running a Kalman filter forward on the data. This requires a covariance estimate for the initial state. On the first iteration of the EM algorithm, this is not a problem. On additional iterations, however, we do not have a valid estimate for the covariance of the smoothed initial estimate. If one were to use the covariance estimate coming out of the Kalman smoother, then this value would decrease every iteration as a result of ‘‘information incest.’’ That is, the initial state would repeatedly get smoothed by using much of the same data as before, but the Kalman

smoother would interpret this data as being “new” and at every iteration the covariance estimate of the initial state would decrease. After enough iterations, the covariance assumed for the initial state will approach zero even though we would not have supreme confidence in the initial state.

A solution to this problem is to forego smoothing the initial state at each step and to use its initial state covariance at every iteration. That is equivalent to taking the initial state out of the set of states \mathbf{X} that are to be estimated by the EM algorithm.

4.2. Precision Problems with the PMHT

Being based thereupon, all versions of the PMHT algorithm suffer the same precision problems that can occur with regular Kalman filter. Verhaegen discusses the source of some of these problems as well as their remedies [63] and such problems are also discussed in most textbooks, such as [4]. The PMHT, however, has a number of its own precision problems that must be taken into account when designing any implementation.

At any step when calculating the posterior association probabilities values for non-clutter targets, if all valid measurements are far from the predicted value $\hat{\mathbf{y}}_{k_r,s}^n(t)$, then it is quite likely that precision limitations will render all of the w s to be zero.⁷ In many such instances, one can forego the use of the w s and assume that there was a missed detection. In comparison with other algorithms, precision is a serious problem in the PMHT, because the non-clutter PDFs in w

4.3. Using Deterministic Annealing

The use of deterministic annealing can both help the PMHT to converge to the global MAP estimate as well as ameliorate precision problems associated with the posterior association probabilities. Which maxima the EM algorithm converges to is highly dependent on the initial state estimates over the entire batch. Deterministic annealing is an approach to reduce the dependence of the algorithm on the initial estimates. However, this may require more iterations of the EM algorithm than if deterministic annealing were not used. As a result, just incorporating deterministic annealing without changing the number of iterations used could theoretically worsen performance.

As shown in (7), the EM function with deterministic annealing replaces $p(\mathbf{K} | \mathbf{X}^n, \mathbf{Z})$ in the regular EM algorithm with

$$p(\mathbf{K} | \mathbf{X}^n, \mathbf{C}^n, \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^n, \mathbf{C}^n)^\beta}{\sum_{\mathbf{K}_1} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^n, \mathbf{C}^n)^\beta d\mathbf{K}_1}. \quad (42)$$

The solution for $p(\mathbf{K} | \mathbf{X}^n, \mathbf{Z})$ given in (15) is a product of w terms. The addition of the β terms in (42) to (15) can take place without explicitly decomposing $p(\mathbf{K} | \mathbf{X}^n, \mathbf{Z})$ into the parts listed in (42).

We note that each w term in (15) has a single value in the numerator as well as a sum of values in the denominator. The inclusion of the β s is done by modifying the w terms as follows. For non-clutter association probabilities, each w shall be adjusted from (17) according to

$$w_{k_r,s(t),r}(t,s) = \frac{(\pi_{k_r,s(t)}(n_t(s),t)c(z_{r,s}^C(t),k_{r,s}(t))\mathcal{N}\{\mathbf{z}_{r,s}(t);\hat{\mathbf{y}}_{k_r,s(t)}(t),\mathbf{R}_{r,s}(t)\})^\beta}{(\pi_0(n_t(s),t)\mu(t,\mathbf{z}_{r,s}(t))c(z_{r,s}^C(t),0))^\beta + \sum_{m=1}^M (\pi_m(n_t(s),t)c(z_{r,s}^C(t),m)\mathcal{N}\{\mathbf{z}_{r,s}(t);\hat{\mathbf{y}}_m(t),\mathbf{R}_{r,s}(t)\})^\beta}. \quad (43)$$

are normally distributed having a covariance equal to that of the measurement. In instances where the process noise covariance is large and the measurement noise covariance is small, precision errors can cause all of the w s for a particular target to be zero much of the time. Thus, paradoxically, the performance of the PMHT can worsen as the magnitude of the measurement covariance decreases. In instances where no measurement noise is present, the PMHT is unusable.

⁷If a clutter model is present, then there will always be a nonzero (clutter) term in the denominator of the w s, thus precision problems can render all of the w s to be zero. However, if there is no clutter term, then precision problems couples with distant measurements can result in the fraction computed for the w to evaluate as 0/0. This can signify that the target was not detected, or that the observation from the target was very far from the predicted position.

Equation (43) is equivalent to raising the numerator and each term in the denominator of (17) to the power of β . The same would be done with the w for the clutter assignment, which we have omitted; there too one would raise the numerator and each term of the denominator to β . Once all of the w terms have been multiplied, as in (15) the result is thus the same as (42) except all common terms have been canceled out between the numerator and denominator. That is, the result is still a numerator raised to β and a denominator consisting of a sum of terms each raised to β . This is the same as the solution given by Wieneke and Koch [66] without derivation and similar to what Strandlie and Zerubia [57] derived.

For precision purposes, the exponentiation of the normal PDFs in (43) is best performed by distributing β 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.

4.4. Sensors with Different Fields of View

The state estimate for the PMHT was derived assuming that all sensors have the same field of view. When the sensors have different fields of view, the calculation of the posterior association probabilities (the w s) is different for each sensor.

Generally, a particular target will have a certain probability of detection when viewed by a particular sensor. In calculating the posterior association probabilities, this detection probability is necessary for calculating the prior association probabilities. This detection probability can be considered to be the product of the probability that the target is located within the field of

5. THE COMPLEXITY OF THE JPDAF VS. THE PMHT

The most complex part of the JPDAF is the evaluation of the posterior association probabilities. These are equivalent to the posterior association probabilities 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 as follows:

$$A_{\text{JPDAF}} = \underbrace{\sum_{l=0}^{\min(n_t, M)} \binom{M}{l}}_{\text{Sum over the number of targets observed}} \underbrace{\binom{n_t}{l}}_{\text{Choose which targets are observed}} \underbrace{\binom{n_t}{l}}_{\text{Choose which measurements are observed}} \underbrace{l!}_{\text{Assign the measurements to the targets}} \quad (44a)$$

$$= {}_2F_0[-n_t, -M; 1]. \quad (44b)$$

view of the sensor times the probability that the sensor detects the target given that it is in its field of view.

As shown in Appendix B, the computation of the prior association probabilities (the π s) is combinatorially complex if all of the targets have different probabilities of detection. This is the case when one takes into account the probability that each target is within the field of view of each sensor. The complexity of this situation may be reduced either by assuming a constant detection probability for all targets within the field of view of a sensor and gating to targets that should be within the field of view given the state estimates. Once gating has been done, this means that the prior and posterior association probabilities for each sensor are calculated assuming a reduced number of targets: only those that fall within the gate for that sensor.

4.5. Out-of-Sequence Measurements

In many practical data fusion schemes, measurements may arrive at the fusion center out of sequence. As was noted by Efe, Ruan, and Willett [18], the PMHT handles such situations with ease. Because the PMHT is a batch algorithm, as long as newly received measurements correspond to a step that has not left the sliding window, the measurements may be added to the batch at any time and are used in the state update.

${}_2F_0$ refers to a generalized hypergeometric function. The step from (44a) to (44b) was performed by noting that the ratio of the a_{l+1} and the a_l th term of the sum in (44a) is:

$$\frac{a_{l+1}}{a_l} = \frac{(l - n_t)(l - M)}{1 + l}. \quad (45)$$

More information on the conversion of sums to hypergeometric functions may be found in [48].

In contrast, although the PMHT allows for more posterior association probabilities 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 measurement association probability w 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 $n_t M$. 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:

$$A_{\text{PMHT}} = IM \sum_{t=2}^N n_t. \quad (46)$$

As shown in Table I, when the number of targets is small, the PMHT will have a higher complexity than

TABLE I
Number of Combinations Considered by the JPDAF versus the PMHT as a Function of the Number of Targets

M	A_{JPDAF}	A_{PMHT}
1	$n_t + 1$	$I \sum_{i=2}^N n_i$
2	$n_t^2 + n_t + 1$	$2I \sum_{i=2}^N n_i$
3	$n_t^3 + 2n_t + 1$	$3I \sum_{i=2}^N n_i$
4	$n_t^4 - 2n_t^3 + 5n_t^2 + 1$	$4I \sum_{i=2}^N n_i$

n_t is the number of measurements in the frame considered, M is the number of targets and I is the number of iterations that the PMHT uses.

the JPDAF. Consistent with the simulation results of Pao [46], practical implementations of the PMHT will never have a lower complexity when there is only one target. However, the complexity of the JPDAF scales exponentially with the number of targets, whereas the complexity of the PMHT scales linearly. So keeping the batch length N fixed, the PMHT has a lower complexity as the number of targets becomes large.

6. USING THE TRACKER OVER TIME

6.1. Growing and Sliding the Batch

The PMHT requires an initial estimate for all of the states in the batch, as well as a the covariance of the state estimate at the first time step. As empirically demonstrated by Willett, Ruan, and Streit [70], a practical, efficient way of running the PMHT using a finite-length batch is by growing and then sliding the batch. In other words, at time $t = 1$ one is given the initial state estimates of the targets $\mathbf{x}_m(1)$. From time $t = 2$ to time $t = N$, where N is the maximum batch-length, the PMHT is run while increasing the batch length by one each time. As mentioned in Section 4.1, in order to eliminate “data incest,” the state estimates at the beginning of the batch should be removed from the estimation that is not updated as part of the Kalman smoothing step. The initial estimates of the states for the rest of the batch are very important for convergence. Even with the use of deterministic annealing, as described in Section 4.3, if the initial state estimates for the batch are particularly bad, then the EM algorithm probably will not converge to the global maximum, nor to a nearby local maximum. For that reason, the best initial state estimates are the estimates from the previous time step. The best initial state estimate for each target at the new step, which was not estimated at the previous time step, is the the best a priori estimate, which is the Kalman filter estimate $\mathbf{x}_m(t | t - 1)$.

At all time steps after the batch length has reached N , the batch should slide one step forward, as demonstrated in Fig. 1 by the Single Shift batch with respect to the Pre-Shift Batch for a length $N = 4$ batch. In this case, the previous estimate from the first time step becomes the new initial state of the first step of the batch, which

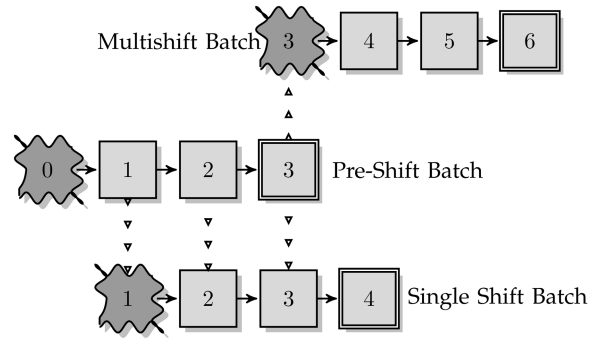


Fig. 1. Different methods of shifting the window for the next time step as shown on a length-4 batch. Each method has its own concerns regarding the consistency of the estimator and the avoidance of “data incest.”

must have an accurate covariance. Section 7 looks at ways of estimating the covariance of this state estimate as well as those of the other state estimates in the batch.

6.2. Other Methods of Sliding the Batch

It should be noted that the aforementioned method of sliding the batch is somewhat *ad-hoc*. Much literature has also focussed on sliding the batch over multiple time steps at once, as shown with the Multishift Batch in Fig. 1. If this approach is taken, one can not obtain a real-time estimate of the target’s location, but must wait until another batch of information has arrived. Ruan, Willett, and Streit [52] suggested that the batches should not overlap by more than one time step and later suggested [70] that the initial state of the slid batch, be it slid one step or many, be calculated using only information equal to or prior to that time period. For example, in the Single-Shift batch in Fig. 1, only information from times 0 and 1 could be used to create the initial state estimate at time 1. However, a length-one PMHT is not a very good tracker. As a result, by using this approach the initial estimates become progressively worse.

The main concern regarding reusing smoothed past state estimates is that it introduces “information incest” in the smoothed state. However, it should be noted that this concern only exists with the initial estimate at the beginning of the batch. The rest of the initial estimates in the batch affect to which local maximum the EM algorithm is likely converge, but they do not affect the location of the maxima in the likelihood function.

The new initial state when using a Single-Shift batch from Fig. 1 and the smoothed state estimate from the previous batch as the initial state, introduces information incest in that it has already been smoothed from future observations. In the Multi-Shift batch, where only a single state overlaps, no information incest is present. However, in the example of Fig. 1, the initial estimate for time steps 4, 5, and 6 in the slid batch would have to be Kalman filter predictions from the estimate at

time 3. If these are far from the true track location, then it is likely that the precision errors will occur, as described in Section 4.2 or that the EM algorithm is likely to converge to a local maximum far from the global maximum.

Thus, there is a tradeoff between how far one slides the batch and how much ‘‘information incest’’ one wishes to allow in the state estimate at the beginning of the slid batch. Wieneke and Koch [66] decided to shift a small number of steps, less than the batch length, and use deterministic annealing. However, in many practical situations, a state estimate is desired at every time step, and thus the method of growing and sliding the batch described in Section 6.1 is a simple approach.

7. COVARIANCE ESTIMATION IN THE PMHT

When sliding the batch, as described in the previous section, an accurate covariance is needed for the new estimate of the first time step of the batch. Additionally, at any time during tracking, one may wish to have a covariance for the target state estimate. Being based upon the EM algorithm, the PMHT does not directly provide this. The simplest approximation is to use the covariance estimates that are produced by the Kalman smoother at each step. However, based upon the Normalized Estimation Error Squared (NEES), which is more closely defined in Section 9, this has been shown to be inconsistent.

In this section, we will look at two methods of producing covariance estimations for the PMHT. One approach, originally presented by Walsh [65], is to use the inverse of the observed information to predict the covariance. A simpler *ad-hoc* approach by Blanding, Willett, Streit, and Dunham [6] is to obtain covariance approximations by normalizing the posterior association and using the covariance estimate from the JPDAF, (as described, for example, in [3]).

7.1. Using the Observed Information to Estimate the Covariance

Letting D be the dimensionality of the state, the observed information matrix is defined as the $DMN \times DMN$ Hessian of the joint likelihood function of the states and observations over all time, evaluated at the state estimate, in this case the EM estimate:

$$I[\hat{\mathbf{X}}, \mathbf{Z}] \triangleq -\nabla_{\hat{\mathbf{X}}}^2 \log p(\mathbf{Z}, \mathbf{X})|_{\mathbf{X}=\hat{\mathbf{X}}} \quad (47)$$

$$= [-\nabla_{\hat{\mathbf{X}}}^2 \log p(\mathbf{X}) - \nabla_{\hat{\mathbf{X}}}^2 \log p(\mathbf{Z} | \mathbf{X})]|_{\mathbf{X}=\hat{\mathbf{X}}} \quad (48)$$

$$= I_{\text{prior}}[\hat{\mathbf{X}}] + I_{\text{data}}[\mathbf{Z} | \hat{\mathbf{X}}]. \quad (49)$$

The inverse of the observed information gives the covariance for all states over all time. The covariances of the individual state estimates are in the $D \times D$ blocks lying on the diagonal of the matrix.

Walsh is the first to have derived these Hessians for the PMHT. We shall give a multisensor adaptation of the observed information matrix as explained by [6], assuming that all observations for a particular sensor at a particular time have the same covariance and measurement matrices. $I_{\text{prior}}[\hat{\mathbf{X}}]$ is given as

$$I_{\text{prior}}[\hat{\mathbf{X}}] = \text{diag} \left[\begin{array}{cc} \chi(t) & -\delta(t) \\ -\delta(t)' & \chi(t+1) \end{array} \right] : \quad t = 1, \dots, N-1 \quad (50)$$

with

$$\chi(t) = \begin{cases} \text{diag}[\mathbf{P}_m(1|1)^{-1} + \mathbf{F}_m(1)'\mathbf{Q}_m(1)^{-1}\mathbf{F}_m(1) : \\ \quad m = 1, \dots, M] & \text{for } t = 1 \\ \text{diag}[\mathbf{Q}_m^{-1}(t-1) + \mathbf{F}_m(t)'\mathbf{Q}_m^{-1}(t)\mathbf{F}_m(t) : \\ \quad m = 1, \dots, M] & \text{for } t = 2, \dots, T-1 \\ \text{diag}[\mathbf{Q}_m^{-1}(T-1) : \\ \quad m = 1, \dots, M] & \text{for } t = T \end{cases} \quad (51)$$

and

$$\delta(t) = \text{diag}[\mathbf{F}_m(t)'\mathbf{Q}_m(t)^{-1} : \\ m = 1, \dots, M] \quad \text{for } t = 1, \dots, T-1. \quad (52)$$

$\chi(t)$ and $\delta(t)$ are $MN \times MN$ matrices. The contribution from the data is given by

$$I_{\text{data}}[\mathbf{Z} | \hat{\mathbf{X}}] = \sum_{s=1}^N (\mathbf{B}(s) - \mathbf{C}(s) + \mathbf{D}(s)), \quad (53)$$

$$\mathbf{B}(s) = \text{diag}[\mathbf{B}(t, s) : t = 1, \dots, T], \quad (54)$$

$$\mathbf{C}(s) = \text{diag}[\mathbf{C}(t, s) : t = 1, \dots, T], \quad (55)$$

$$\mathbf{D}(s) = \text{diag}[\mathbf{D}(t, s) : t = 1, \dots, T], \quad (56)$$

$$\mathbf{B}(t, s) = \text{diag}[\mathbf{H}_{m,s}(t)'\tilde{\mathbf{R}}_{m,s}(t)^{-1}\mathbf{H}_{m,s}(t) : \\ m = 1, \dots, M], \quad (57)$$

$$\mathbf{C}(t, s) = \text{diag} \left[\mathbf{H}_s(t)'\mathbf{R}_s(t)^{-1} \right. \\ \left. \times \left(\sum_{r=1}^{n_r(s)} w_{m,r}(t, s) \nu_{m,r}(s, t) \nu_{m,r}(s, t)' \right) \right. \\ \left. \times \mathbf{R}_s(t)^{-1}\mathbf{H}_s(t) : m = 1, \dots, M \right], \quad (58)$$

$$\mathbf{D}(t, s) = \sum_{r=1}^{n_r(s)} \mathbf{D}_r(t, s)\mathbf{D}_r(t, s)', \quad (59)$$

and

$$\mathbf{D}_r(t, s) = \begin{bmatrix} w_{m,r}(t, s)\mathbf{H}_s(t)'\mathbf{R}_1^{-1}(t)\nu_{1,r}(t) \\ \vdots \\ w_{M,r}(t, s)\mathbf{H}_s(t)'\mathbf{R}_M^{-1}(t)\nu_{M,r}(t) \end{bmatrix}. \quad (60)$$

Note that $B(t, s)$ contains the synthetic measurement covariance $\tilde{\mathbf{R}}_{m,s}(t)$, whereas the other equations only contain the covariance of the measurements. The innovations are defined as follows:

$$\nu_{m,r}(s, t) = \mathbf{H}_s(t)\mathbf{x}_m(t) - \mathbf{z}_{r,s}(t). \quad (61)$$

The derivation of the observed information was performed under the assumption that the transition matrix \mathbf{F} and the process noise covariance matrix \mathbf{Q} are invertible. This is true when using the discretized continuous white noise acceleration model for the motion, but is not true when using the discrete white noise acceleration model [4], in which case a pseudoinverse would be necessary.⁸

7.2. A Simpler, *Ad-hoc* Approach to Covariance Estimation

We shall extend the *ad-hoc* covariance estimation approach taken by Blanding, Willett, Streit, and Dunham [6] to the multisensor case. That is, we shall show that estimator consistency is improved when the posterior assignment probabilities are normalized and the covariance estimate from the MSJPDAF is used. The MSJPDAF is a generalization of the JPDAF to multiple sensors (see, for example [3] 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 [45]. 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 posterior association probabilities sum to one over all measurements for a particular target plus the probability that target was not detected. This is because the MSJPDAF does not make the same assumption as the PMHT, that each target can produce more than one measurement at a particular time. By 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 was not detected by a particular sensor s at a particular time t is given as follows:

$$\beta_{m,0,s}(t) = \prod_{r=1}^{n_r} (1 - w_{m,r}(t, s)). \quad (62)$$

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

$$\beta_{m,r,s}(t) = w_{m,r}(t, s) \frac{1 - \beta_{m,0,s}(t)}{\sum_{r=1}^{n_r} w_{m,r}(t, s)}. \quad (63)$$

⁸The Moore-Penrose pseudoinverse of matrix A is $A^{-1} = (AA')^{-1}A$. If AA' is poorly conditioned, then the pseudoinverse can produce bad results. In simulations using the discrete white noise acceleration model, we have observed that conditioning is often a problem in evaluating the observed inverse of the information matrix.

The covariance update from the parallel MSJPDAF is:

$$\mathbf{P}_m(t | t) = \left(\sum_{\mathcal{C}} \beta_{m,\mathcal{C}}(t) (\mathbf{P}_{m,\mathcal{C}}(t | t) + \mathbf{x}_{m,\mathcal{C}}(t | t) \mathbf{x}_{m,\mathcal{C}}(t | t)') \right) - \mathbf{x}_m(t | t) \mathbf{x}_m(t | t)'. \quad (64a)$$

Equation (64a) is the form given by Pao and Frei [45]. \mathcal{C} represents a particular combination of assignments between sensors for a particular target. Each $\beta_{m,\mathcal{C}}$ 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 \mathcal{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 (64a), $\mathbf{x}_{m,\mathcal{C}}(t | t)$ represents the state update of the Kalman filter if the measurement-assignment for all sensors given by \mathcal{C} is correct. This means fusing the actual observations in the same way that the synthetic measurements were fused in (29), (30), and (31) and then updating the state estimate from the PMHT. By updating the PMHT state estimate reusing the observations, a certain degree of “data incest” is added, but again this method is *ad-hoc* and problems with the “incest” were not observed in previous work using a single sensor [6]. The covariance $\mathbf{P}_{m,\mathcal{C}}(t | t)$ is likewise what the covariance would be if assignment \mathcal{C} is correct. Since only the first state of the PMHT has a covariance, the pre-update covariance at time t would consist of $\mathbf{P}_m(t | t - 1)$, that is the forward predicted covariance from the previous estimate. In this manner, this covariance estimation algorithm must be done in order from the first to the last state estimate. $\mathbf{x}_m(t | t)$ is a the weighted average of these other state updates:

$$\mathbf{x}_m(t | t) = \sum_{\mathcal{C}} \beta_{m,\mathcal{C}} \mathbf{x}_{m,\mathcal{C}}(t | t). \quad (65)$$

8. A SUMMARY OF THE BASIC ALGORITHM

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 \mathbf{H} . If the detection probabilities are different for all targets, then the prior and posterior association probabilities referenced from the appendices should be used. If the observations have different measurement matrices, as might be the case if range-rate information is available, then the alternative state update equations given in Section 4 should be used.

- 1) Set the initial state estimate for each target at the current time step to the Kalman filter predicted value of the state $\mathbf{x}_m(t | t - 1)$.
- 2) For each sensor and observation, calculate the posterior assignment probabilities $w_{k_{r,s}(t),r}(t, s)$ according to (18) and (19).

- 3) Create the synthetic measurements $\tilde{\mathbf{z}}_{m,s}(t)$ with their corresponding measurement covariances $\tilde{\mathbf{R}}_{m,s}(t)$ for each target, measurement and sensor according to (25) and (26) or (35) and (34).
- 4) Merge the synthetic measurements between sensors according to (32) and (33).
- 5) Using the fixed initial state estimate $\mathbf{x}_m(1)$ and state covariance estimate $\mathbf{P}_m(1)$ for each track, run the Kalman smoother, as described in Appendix D using the merged measurements as the observations. Do not smooth the initial state.
- 6) Update the confusion matrix using (41).
- 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 (64a) or as described in Section 7.1.
- 9) Slide the batch window forward, using the proper covariance estimate for the new initial state, such as in (64a) or as described in Section 7.1.
- 10) Go to 1.

9. SIMULATION

We compared the consistency and track retention rates of the MSJPDAF and the MSPMHT with and without deterministic annealing and using the various covariance estimation methods of Section 7 when using two sensors and two targets. The sequential version of the MSJPDAF was used.⁹ We used the two-dimensional discretized continuous white noise acceleration model.¹⁰ Both sensors had the same field of view and measured in Cartesian coordinates without classification information. The ordering of the elements of the state was $[x, y, \dot{x}, \dot{y}]$. Using an 80% probability of detecting each track at each sensor, the simulation parameters were

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (66)$$

$$\mathbf{Q} = \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix} \sigma_p^2, \quad (67)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (68)$$

⁹[47] discusses the sequential MSJPDAF algorithm, but provides an incorrect state covariance estimate. [32] provides the correct state covariance estimate when solving a different problem.

¹⁰The one dimensional version is described in [4]; the two-dimensional version follows from it.

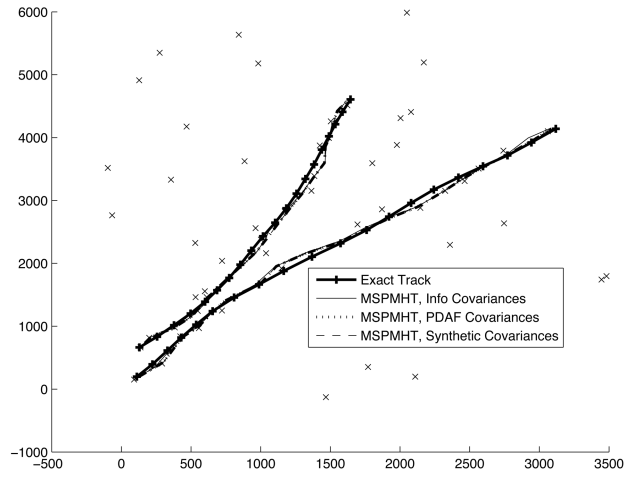


Fig. 2. A typical run of the simulation. The observations from the first sensor and the last frame of clutter for that sensor are shown.

and

$$\mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_m^2. \quad (69)$$

The sampling time T was set to 30 seconds. 1,000 Monte Carlo Runs were performed. The first target was placed at the origin given an initial velocity of 7 m/s at a 59° angle from the x axis. The second target was assigned the same speed, and was placed at 500 m on the y axis. Ascending at a 52° degree angle from the x axis. σ_m was chosen to be 50 m for both sensors and σ_p $0.1 \text{ m}^2/\text{s}^3$. Clutter was generated uniformly in a viewing rectangle bounded between $(-200 \text{ m}, -200 \text{ m})$ and $(3.5 \text{ km}, 6 \text{ km})$. The number of clutter points was determined at each time step by a Poisson random variable having mean 23. The MSPDAF was gated to observations within a 99.97 percent probability region around the estimated location of the target. The simulation was initialized by giving two correctly assigned measurements for each track to information filters (the information filter is described in [4]). For the PMHT, a window growing to a maximum of length 10 was used. After the 10th time step, the window slid. The PMHT was performed using 10 iterations at each step.

Fig. 2 shows a typical run. The proportion of tracks not lost at each step was calculated. 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. Fig. 3 shows the track-loss performance. As expected, deterministic annealing significantly improved the track-loss performance of the MSPMHT.

To evaluate the consistency of the trackers, the average normalized estimation error squared (NEES) for tracks that were not lost was calculated and averaged over all tracks and Monte Carlo runs. The NEES is defined as

$$\text{NEES} = (\mathbf{x}(t) - \mathbf{x}(t|t))\mathbf{P}(t|t)^{-1}(\mathbf{x}(t) - \mathbf{x}(t|t))^T. \quad (70)$$

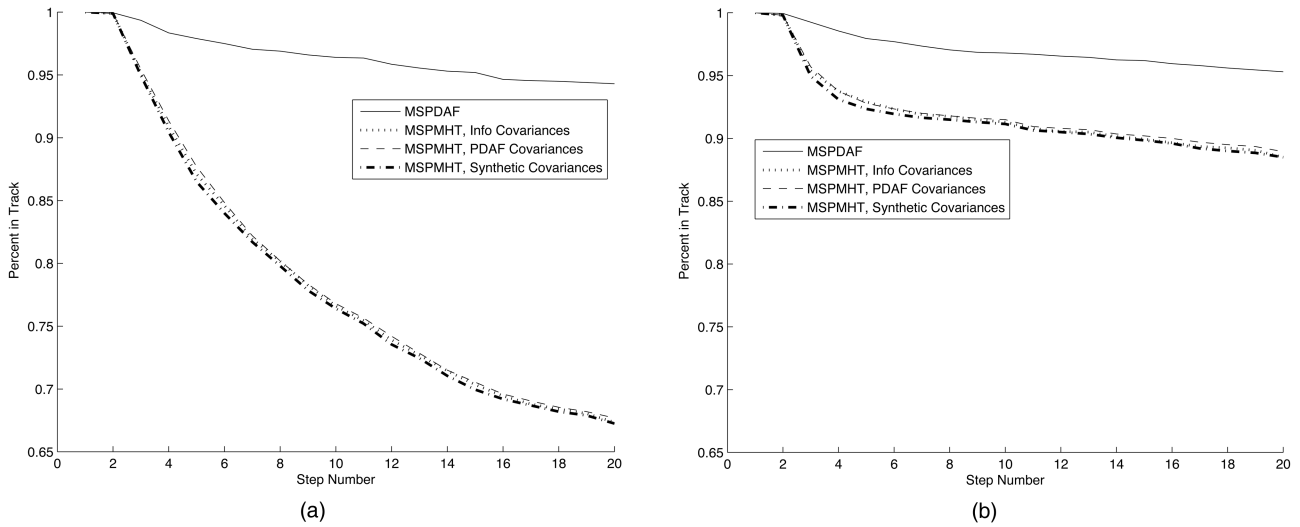


Fig. 3. The fraction of tracks not lost at each step shown with and without the use of deterministic annealing. (a) Without DA. (b) With DA.

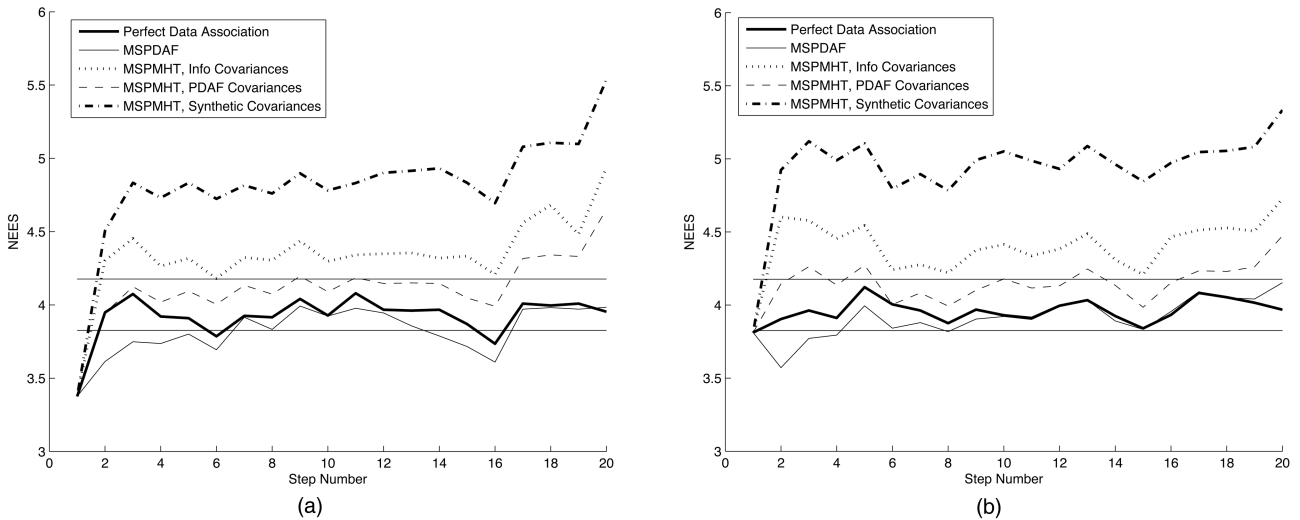


Fig. 4. The average NEES for tracks not lost by any of the trackers. The horizontal lines mark the 95 percent confidence interval of the NEES. (a) Without DA. (b) With DA.

The NEES for a particular track is a random variable with 4 degrees of freedom. The average NEES over n Monte Carlo Runs is a chi-squared random variable with $4n$ degrees of freedom. As mentioned in [4], the inverse Cumulative Distribution Function (CDF) of a chi-squared random variable given ν degrees of freedom, where $\nu > 100$, is approximately

$$\chi_{\nu}^2(p) \approx \frac{1}{2} \left(\mathcal{G}(p) + \sqrt{2\nu - 1} \right)^2. \quad (71)$$

p is the probability at which the inverse CDF is to be evaluated. $\mathcal{G}(p)$ is the inverse CDF of the standard normal distribution. Thus for the 95 percent confidence interval we get:

$$[\mathcal{G}(0.025), \mathcal{G}(0.975)] = [-1.96, 1.96]. \quad (72)$$

As can be seen in Fig. 4, the use of MSJPDFAF covariances in the two-sensor MSPMHT improves the consistency of the tracker, even outperforming those ob-

tained by using the observed information approximations. However, as can be seen in Fig. 3, the use of improved covariances has little effect on track retention. The apparent inconsistency of the initial estimate in Fig. 4 stems from the fact that those tracks most often lost by the PMHT were those where the covariance of the initial estimate was underestimated. Deterministic annealing significantly improves track retention. This coincides with previous results done using a single sensor on a single track [57] and [66].

All together, the MSPMHT performs worse than the MSJPDFAF in the two sensor scenario. However, the poor performance of the basic PMHT has been shown in previous literature and was one of the motivations for the creation of other PMHT algorithms, such as the MF-PMHT by Blanding, Willett, Streit, and Dunham [7]. The multisensor PMHT presented in this paper can form the basis of such modified algorithms.

10. 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. We demonstrated that deterministic annealing can significantly improve tracker performance and we showed that the JPDAF covariance approximations provide the most consistent covariance estimates in the simulation, but this consistency has little effect on the track retention as compared to using the synthetic covariances. We also provided a simplified solution for the prior association probabilities (the π s). 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, such as the Turbo PMHT and the MFPMHT, which achieve better track retention.

APPENDIX A. A DERIVATION OF THE POSTERIOR ASSOCIATION PROBABILITIES $w_{l,r}(t,s)$

The derivation of the posterior association probabilities for the PMHT is dependent upon the sensor in question. For simplicity, we shall assume that we are only considering that which is seen by sensor s and we shall suppress s from the notation. As has typically been done, we shall derive it assuming that all sensors see everything and that there is no gating.

In order to highlight the complexity reduction of the PMHT measurement model, we shall begin by assuming the regular target-measurement assignment model (i.e., that a target can produce only one observation per sensor at each time) before finishing the solution using the PMHT measurement model. Let there be a total of M non-clutter targets and one clutter target $m = 0$. Let $\mathbf{X}(t)$ be the state of the Kalman filters for all non-clutter targets at time t . Define $\mathbf{z}_r(t) \in \mathbf{Z}(t)$ to be the r th measurement out of the set of $\mathbf{Z}(t)$ measurements at time t , which consists of n_t measurements (in order to simplify the notation, we shall omit the subscript on n). $z_r^C(t)$ shall be the classification value associated with measurement $z_r(t)$. Define $k_r(t)$ to be source of the r th measurement at time t . \mathbf{C} shall be the set of all classification probabilities $c(i,m) = \Pr(z_r^C(t) = i | k_r(t) = m)$. The classification of the target is assumed independent of time and the target state. The probability that measurement z_r at time t came from target m given the current set of observations, the set of classification probabilities, and the estimated state of the set of Kalman filter is $w_{m,r}(t) = \Pr(k_r(t) = m | \mathbf{X}(t), \mathbf{Z}(t), \mathbf{C}, n)$. Using Bayes Rule, this may be decomposed as follows:

$$w_{m,r}(t) = \frac{\Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), \mathbf{C}, n) \Pr(k_r(t) = m | \mathbf{X}(t), \mathbf{C}, n)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) | k_r(t) = p, \mathbf{X}(t), \mathbf{C}, n) \Pr(k_r(t) = p | \mathbf{X}(t), \mathbf{C}, n)} \quad (73a)$$

$$= \frac{\Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), n) \pi_m(n) c(z_r^C, m)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) | k_r(t) = p, \mathbf{X}(t), n) \pi_p(n) c(z_r^C, p)} \quad (73b)$$

Once we fix $k_r(t) = m$, the probability of observing z_r is independent of the other track-measurement associations. Therefore, we may decompose $\Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), n)$ into two parts:

$$\begin{aligned} \Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), n) \\ = \Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_m(t), n) \cdot \Pr(z_r(t) | k_r(t) = m, x_m(t), n). \end{aligned} \quad (74)$$

$\mathbf{Z}(t) \setminus z_r(t)$ represents the set $\mathbf{Z}(t)$ without measurement $z_r(t)$. $\mathbf{X}(t) \setminus x_m(t)$ represents the set $\mathbf{X}(t)$ without the elements corresponding to track m . Let $f_{t,m}(q)$ be the PDF of the estimated target location at time t for target m given $\mathbf{X}(t)$. If target m is not clutter, then this is the Kalman filter estimate, which is normally distributed. This normal distribution comes directly from the Kalman filter in (9) and has a covariance $\mathbf{R}_r(t)$ equal to that associated with the measurement. If target m is clutter, i.e., $m = 0$, then $f_{t,0}(q)$ is the PDF of the clutter at point q . Usually this is assumed to be uniformly distributed over the field of view, but we shall designate it by $\mu(t, q)$ to allow for the use of a generic continuous distribution to be used. We shall assume that $\mu(t, q)$ is continuous as a function of q . Hence we obtain

$$f_{t,m}(q) = \begin{cases} \mu(t, q) & m = 0 \\ \mathcal{N}(q; \mathbf{H}_r \mathbf{X}(t), \mathbf{R}_r(t)) & m \neq 0 \end{cases} \quad (75)$$

Because the PDFs of the measurements coming from targets and those originating from clutter are assumed continuous, $\Pr(z_r | k_r(t) = m, \mathbf{X}(t), n)$ may be expressed as the probability of the observation being within a certain region around the observation as the size of that region approaches zero. We shall denote the size of this region as Δ and the region itself, which is centered about the observation z_r as $\Delta(z_r)$. Formulating this probability as a limit allows us to deal with zeros in the numerator and denominator of (74).

$$\Pr(z_r | k_r(t) = m, \mathbf{X}(t), n) = \lim_{\Delta \rightarrow 0} \int_{q \in \Delta(z_r)} f_{t,m}(q) dq \quad (76)$$

$$= \lim_{\Delta \rightarrow 0} f_{t,m}(z_r) \Delta. \quad (77)$$

Substituting (77) and (74) into (73b) we get

$$w_{m,r}(t) = \lim_{\Delta \rightarrow 0} \frac{\Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_m(t), n) \cdot f_{t,m}(z_r) \cdot \Delta \pi_m(n) \cdot c(z_r^C, m)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_p(t), n) f_{t,p}(z_r) \Delta \pi_p(n) c(z_r^C, p)} \quad (78a)$$

$$= \frac{\Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_m(t), n) \cdot f_{t,m}(z_r) \cdot \pi_m(n) \cdot c(z_r^C, m)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_p(t), n) f_{t,p}(z_r) \pi_p(n) c(z_r^C, p)} \quad (78b)$$

The jump from (78a) to (78b) was accomplished by noting that Δ could be factored out of the sums and products and thus cancels in the numerator and denominator.

Equation (78b) is the solution assuming that each target can produce only a single measurement. The evaluation of $\Pr(\mathbf{Z}(t) \setminus z_r(t) \mid \mathbf{X}(t) \setminus x_m(t), n)$ is combinatorially complex. However, under the PMHT measurement model, whereby a single target can produce any number of measurements at one time, this becomes simpler. Under the PMHT measurement model, (74) simplifies as follows:

$$\begin{aligned} \Pr(\mathbf{Z}(t) \mid k_r(t) = m, \mathbf{X}(t), n) \\ = \Pr(\mathbf{Z}(t) \setminus z_r(t) \mid \mathbf{X}(t), n) \cdot \Pr(z_r(t) \mid k_r(t) = m, x_m(t), n). \end{aligned} \quad (79)$$

This thus leaves a common term in (78b) that can be cancelled, giving us:

$$w_{m,r}(t) = \frac{f_{t,m}(z_r) \pi_m(n) c(z_r^C, m)}{\sum_{p=0}^M f_{t,p}(z_r) \pi_p(n) c(z_r^C, p)}. \quad (80)$$

Substituting the appropriate distributions for $f_{t,m}(z_r)$ and $f_{t,p}(z_r)$ gives us the form given in (17).

APPENDIX B. A DERIVATION OF THE PRIOR ASSOCIATION PROBABILITIES

$$\pi_m(n_r(s), t)$$

B.1. A General Derivation of the Prior Association Probabilities

Adopting the notation from the previous section, the prior association probabilities are defined as

$$\pi_m(n) = \Pr(k_r(t) = m \mid \mathbf{X}(t), \mathbf{C}, n). \quad (81)$$

In this case, we are suppressing the conditioning on $\mathbf{X}(t)$ and \mathbf{C} in the notation of π , because, assuming that the clutter is uniformly distributed in the viewing area, the actual location of the target has no bearing on the solution. In this section, we shall also suppress the conditioning on time in order to simplify notation. We shall derive the prior association probabilities assuming that no gating is taking place and that all sensors have the same field of view. It should be noted that the derivation of the π values are the embodiment of the PMHT measurement model. However, the PMHT measurement model is only an approximation. For this purpose we shall derive the π s based upon the usually more realistic model that a single sensor can only observe at most one measurement of each target at each time step.

The derivation of the π s takes place under the constraint that each measurement came either from a target or from clutter, that is:

$$\sum_{m=0}^M \pi_m(n) = 1. \quad (82)$$

Thus, solving for the value of the non-clutter π is sufficient to tell us the value of the clutter π for a particular n . Let us now determine the values of $\pi_m(n)$

given that $m > 0$. Using the Law of Total Probability to perform a decomposition based upon the number n_o , of targets observed we may thus write:

$$\pi_m(n) \triangleq \Pr(k_r = m \mid n) \quad (83a)$$

$$= \sum_{k=1}^{\min(n,M)} \Pr(k_r = m \mid n, n_o = k) \Pr(n_o = k \mid n). \quad (83b)$$

The Law of Total Probability may be used to simplify the first term of (83b) by adding conditioning upon whether observation z_r originated from a target. The first term of (83b) may be simplified as follows:

$$\begin{aligned} \Pr(k_r = m \mid n, n_o = k) \\ = \Pr(k_r = m \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \\ \times \Pr(\mathbf{z}_r \in \mathbb{M} \mid n, n_o = k). \end{aligned} \quad (84)$$

The second term in (84) can be found by counting: if k targets are observed, then the probability that a particular observation is a target is simply the ratio of the number of targets observed to the total number of observations. We thus have

$$\Pr(\mathbf{z}_r \in \mathbb{M} \mid n, n_o = k) = \frac{k}{n}. \quad (85)$$

The first term of (84) may be decomposed using the Law of Total Probability:

$$\begin{aligned} \Pr(k_r = m \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \\ = \Pr(k_r = m \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}, m \text{ observed}) \\ \times \Pr(m \text{ observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \quad (86a) \\ = \left(\frac{1}{k}\right) \Pr(m \text{ observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}). \end{aligned} \quad (86b)$$

The notation “ m observed” in the conditioning is an abbreviation for “the m th target was observed.” The jump from (86a) to (86b) was done by noting that if we know that target m was observed, observation r is a target and that k targets were observed, then by counting, we know that the association probability is $1/k$.

Let us define some additional notation. Let \mathbb{M} be the set of all measurements originating from a target, and p_m be the probability of detecting target m on a particular scan. There shall be no p_0 for clutter. We shall designate the set of all p_m as \mathbb{P}_d . There are $\binom{M}{k}$ ways of choosing which k targets are observed for each item in the sum. Let $\mathbb{P}_d(k)$ be the set of all products of k -combinations from \mathbb{P}_d without repetition. For example, if $k = 2$ and $M = 3$, then $\mathbb{P}_d(2) = \{p_1 p_2, p_1 p_3, p_2 p_3\}$ (the specific ordering of the terms is not important). Define $e_k(y)$ to be an enumerating function over \mathbb{P}_k , giving us the y th ordered element from $\mathbb{P}_d(k)$. $\mathbb{M}_k(y)$ shall represent the set of k targets whose detection probabilities are part of

$e_k(y)$, i.e., it is a set of observed targets. Let $\bar{e}_k(y)$ be, if $k < M$, the product of all $M - k$ elements of \mathbb{P}_d that are not used in $e_k(y)$, or $\bar{e}_k(y) = 1$ if $k = M$. For example, for $k = 2$ and $M = 3$, then $e_2(y)$ for $y = 1$ is $p_1 p_2$, for $y = 2$ is $p_1 p_3$, and for $y = 3$ is $p_2 p_3$. Likewise $\bar{e}_2(y)$ for each y is respectively $1 - p_3$, $1 - p_2$, and $1 - p_1$. Let $I(x)$ be an indicator function that is 1 if x is nonzero.

Equation (86b) is equal to the sum of the probabilities of all combinations of k observed targets such that the m th target is observed:

$$\begin{aligned} \Pr(m \text{ observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \\ = \frac{\sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) I(m \in \mathbb{M}_k(y))}{\sum_{l=1}^{\binom{M}{k}} e_k(l) \bar{e}_k(l)}. \end{aligned} \quad (87)$$

Combining (87) with (86b) and (85) to form (84), we get the following solution to the first term of (83b):

$$\begin{aligned} \Pr(k_r = m \mid n, n_o = k) \\ = \left(\frac{\sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) I(m \in \mathbb{M}_k(y))}{\sum_{l=1}^{\binom{M}{k}} e_k(l) \bar{e}_k(l)} \right) \frac{1}{n}. \end{aligned} \quad (88)$$

The second term of (83b) may be simplified using Bayes' Theorem:

$$\Pr(n_0 = k \mid n) = \frac{\Pr(n \mid n_0 = k) \Pr(n_0 = k)}{\sum_{j=0}^{\min(n, M)} \Pr(n \mid n_0 = j) \Pr(n_0 = j)}. \quad (89)$$

$\Pr(n \mid n_0 = k)$ from (89) is the probability that there are $n - k$ clutter points. We shall designate this probability by the function $\xi(n - k)$. $\Pr(n_0 = k)$ from (89) is the probability that k targets are observed and may be written as follows:

$$\Pr(n_0 = k) = \sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y). \quad (90)$$

Substituting (90) into (89), we get

$$\Pr(n_0 = k \mid n) = \frac{\xi(n - k) \left(\sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) \right)}{\sum_{i=0}^{\min(n, M)} \xi(n - i) \left(\sum_{y=1}^{\binom{M}{i}} e_i(y) \bar{e}_i(y) \right)}. \quad (91)$$

Substituting (91) and (88) back into (83b), we get the following expression for $\pi_m(n)$:

$$\pi_m(n) = \begin{cases} 1 - \sum_{j=1}^M \pi_j(n) & m = 0 \\ \frac{\sum_{k=1}^{\min(n, M)} \xi(n - k) \sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) I(m \in \mathbb{M}_k(y))}{n \sum_{i=0}^{\min(n, M)} \xi(n - i) \left(\sum_{y=1}^{\binom{M}{i}} e_i(y) \bar{e}_i(y) \right)} & m \neq 0 \end{cases} \quad (92)$$

B.2. Simplifying the Prior Association Probabilities

Typically, the detection probability of the targets is unknown a priori. In this case, it is often simplest to assume that all of the targets have the same detection probability P_D , which would be chosen based upon the properties of the sensor and typical targets. When that is the case, $\Pr(n_0 = k)$ from (90) is the same as the probability of having k successes out of M Bernoulli trials each with a success probability of P_D :

$$\Pr(n_0 = k) = \binom{M}{k} P_D^k (1 - P_D)^{M-k}. \quad (93)$$

Additionally, $\Pr(m \text{ Observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M})$ from (86b) may easily be solved by counting. With M targets total and k targets observed, the probability of observing any particular target is just k/M . Hence we obtain

$$\Pr(m \text{ Observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) = \frac{k}{M}. \quad (94)$$

Using these simplifications, $\pi_m(n)$ is the same for all $m \neq 0$ and (92) may be written in the following simplified form:

$$\pi_m(n) = \begin{cases} 1 - M\pi_1(n) & m = 0 \\ \frac{\sum_{k=1}^{\min(n, M)} k \xi(n - k) \binom{M}{k} P_D^k (1 - P_D)^{M-k}}{Mn \sum_{i=0}^{\min(n, M)} \xi(n - i) \binom{M}{i} P_D^i (1 - P_D)^{M-i}} & m \neq 0 \end{cases}. \quad (95)$$

The probability $\xi(k)$ of observing k clutter points at time t is often modeled as a Poisson probability mass function with mean λV where λ represents the mean amount of clutter per unit volume and V is the volume of the viewing area. Thus, we have

$$\xi(k) = \frac{(\lambda V)^k}{k!} e^{-\lambda V}. \quad (96)$$

Substituting (96) into (95) for $m \neq 0$, we get

$$\pi_m(n) \Big|_{m \neq 0} = \frac{\sum_{k=1}^{\min(n, M)} \frac{k}{(\lambda V)^k (n - k)!} \binom{M}{k} P_D^k (1 - P_D)^{M-k}}{Mn \sum_{i=0}^{\min(n, M)} \frac{1}{(\lambda V)^i (n - i)!} \binom{M}{i} P_D^i (1 - P_D)^{M-i}}. \quad (97)$$

It can be noted that the ratio of consecutive terms of the sums in the numerator and denominator of (97) are ratios of polynomials in k . That is, the ratio of the a_{k+1} and the a_k th term of the sum in the numerator is

$$\frac{a_{k+1}}{a_k} = \frac{(k - M)(k - n)}{k} \left(\frac{P_D}{(1 - P_D)\lambda V} \right). \quad (98)$$

Likewise the ratio of the a_{k+1} and the a_k th term of the sum in the denominator is

$$\frac{a_{k+1}}{a_k} = \frac{(k - M)(k - n)}{k(k + 1)} \left(\frac{P_D}{(1 - P_D)\lambda V} \right). \quad (99)$$

TABLE II

Examples of $\pi_m(n)|_{m \neq 0}$ and $\bar{\pi}$ when all Targets have the Same Detection Probability and a Poisson Clutter Model is Used

M	$\pi_m(n) _{m \neq 0}$	$\bar{\pi}$
1	$\frac{P_D}{nP_D + (1 - P_D)\lambda V}$	$\left(\frac{1}{P_D} - 1\right) + n - 1$
2	$\frac{P_D^2(n - 1 - \lambda V) + P_D \lambda V}{P_D^2 n(n - 1) + 2nP_D \lambda V(1 - P_D) + (1 - P_D)^2 \lambda^2 V^2}$	$\frac{P_D(n - 1)}{P_D(1 - n + \lambda V) - \lambda V} + \left(\frac{1}{P_D} - 1\right) + n - 1$

Thus, following the method in [48], the sums may be rewritten in terms of hypergeometric functions

$$\pi_m(n) = \begin{cases} 1 - M\pi_1(n) & m = 0 \\ -\frac{{}_2F_0\left[1 - M, 1 - n; \frac{P_D}{(1 - P_D)\lambda V}\right]}{{}_2F_0\left[-M, -n; \frac{P_D}{(1 - P_D)\lambda V}\right]} & m \neq 0 \end{cases} \quad (100)$$

The function ${}_2F_0(a_1, a_2; z)$ in (100) is a hypergeometric function.

Table II shows $\pi_m(n)|_{m \neq 0}$ from (100) for one and two targets. Previous publications, such as [3], have also derived (100) for a single target under a Poisson clutter model.

APPENDIX C. FURTHER SIMPLIFICATIONS TO THE w s

If the probability of detection is the same for all targets, then the π s and the w s may be simplified one step further. Examining (95), it can be seen that all of the π values are the same for all targets. Using the simplified form of the w s in equation (17), we can divide the π term from the numerator into the denominator to get

$$w_{k_{r,s}(t),r}(t,s) = \frac{\mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\}c(z_{r,s}^C(t), k_{r,s}(t))}{\bar{\pi}(n_t(s), t)\mu(t, \mathbf{z}_{r,s}(t))c(z_{r,s}^C(t), 0) + \sum_{m=1}^M \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\}c(z_{r,s}^C(t), m)} \quad (101)$$

This lowers the total number of multiplications needed and it is more efficient to calculate $\bar{\pi}$ than to calculate the π values for clutter and targets separately. When a Poisson clutter model is used, as is the case in (100), then, maintaining the notation from Section B, $\bar{\pi}$ is given as follows:

$$\bar{\pi} = \frac{\pi_0}{\pi_{m \neq 0}} \quad (102a)$$

$$= -M - \frac{{}_2F_0\left[-M, -n; \frac{P_D}{(1 - P_D)\lambda V}\right]}{{}_2F_0\left[1 - M, 1 - n; \frac{P_D}{(1 - P_D)\lambda V}\right]} \quad (102b)$$

Some expressions for $\bar{\pi}$ as a function of the number of targets are given in Table II.

APPENDIX D. THE KALMAN SMOOTHER

In this section, we summarize the equations for the Kalman smoother, as described in [4]. $\mathbf{x}(t_1 | t_2)$ represents the state estimate at time t_1 given all observations from time 1 to t_2 , whereby $\mathbf{x}(0 | 0)$ is the initial estimate. $\mathbf{P}(t_1 | t_2)$ is the covariance of the aforementioned state estimate. $\mathbf{F}(t)$ is the state transition matrix and $\mathbf{H}(t)$ the measurement matrix, as shown in the dynamic equations in (8) and (9). In keeping with the dynamic model, $\mathbf{Q}(t)$ is the process noise covariance and $\mathbf{R}(t)$ the measurement noise covariance, the noises being zero-mean and white. $\mathbf{z}(t)$ is the observation at time t and $\mathbf{R}(t)$ is its covariance. The Kalman smoother consists of a Kalman filtering step, after which a smoothing step is applied. Letting \mathbf{I} be the identity matrix, the Kalman filter equations to calculate $\mathbf{x}(t | t)$ from $\mathbf{x}(t - 1 | t - 1)$ and the observations at time t are given by

$$\mathbf{x}(t | t - 1) = \mathbf{F}(t)\mathbf{x}(t - 1 | t - 1), \quad (103)$$

$$\hat{\mathbf{y}}(t) = \mathbf{H}(t)\mathbf{x}(t | t - 1), \quad (104)$$

$$\mathbf{P}(t | t - 1) = \mathbf{F}(t)\mathbf{P}(t - 1 | t - 1)\mathbf{F}(t)' + \mathbf{Q}(t), \quad (105)$$

$$\mathbf{W}(t) = \mathbf{P}(t | t - 1)\mathbf{H}(t)'[\mathbf{R}(t) + \mathbf{H}(t)\mathbf{P}(t | t - 1)\mathbf{H}(t)']^{-1}, \quad (106)$$

$$\mathbf{P}(t | t) = [\mathbf{I} - \mathbf{W}(t)\mathbf{H}(t)]\mathbf{P}(t | t - 1)[\mathbf{I} - \mathbf{W}(t)\mathbf{H}(t)]' + \mathbf{W}(t)\mathbf{R}(t)\mathbf{W}(t), \quad (107)$$

and

$$\mathbf{x}(t | t) = \mathbf{x}(t | t - 1) + \mathbf{W}(t)[\mathbf{z}(t) - \hat{\mathbf{y}}(t)]. \quad (108)$$

Assuming that there are N time-periods of data available, the smoothing is performed by starting with the final estimate $\mathbf{x}(N | N)$ and going backwards along the previous estimates, smoothing them using the following

equations

$$\mathbf{C}(t) = \mathbf{P}(t | t)\mathbf{F}(t)\mathbf{P}(t + 1 | t)^{-1}, \quad (109)$$

$$\mathbf{x}(t | N) = \mathbf{x}(t | t) + \mathbf{C}(t)[\mathbf{x}(t + 1 | N) - \mathbf{x}(t + 1 | t)], \quad (110)$$

and

$$\mathbf{P}(t | N) = \mathbf{P}(t | t) + \mathbf{C}(t)[\mathbf{P}(t + 1 | N) - \mathbf{P}(t + 1 | t)]\mathbf{C}(t)'. \quad (111)$$

REFERENCES

- [1] K. M. Alexiev
Multiple target tracking using Hough transform PMHT algorithm.
In *First International IEEE Symposium on "Intelligent Systems"*, vol. 1, 2002, 227–232.
- [2] D. Avitzour
A maximum likelihood approach to data association.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 2 (Apr. 1992), 560–566.
- [3] Y. Bar-Shalom and X-R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
Storrs, CT: YBS Publishing, 1995.
- [4] Y. Bar-Shalom, X-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York: John Wiley & Sons, 2001.
- [5] D. P. Bertsekas
Nonlinear Programming (2nd ed.).
Belmont, MA.: Athena Scientific, 2003, 107–108.
- [6] W. R. Blanding, P. Willett, R. L. Streit, and D. Dunham
Consistent covariance estimation for PMHT.
In *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 6699, San Diego, CA, Aug. 2007, 66 990I-1–66 990I-11.
- [7] W. R. Blanding, P. Willett, R. L. Streit, and D. Dunham
Multi-frame assignment PMHT that accounts for missed detections.
In *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007.
- [8] R. A. Boyles
On the convergence of the EM algorithm.
Journal of the Royal Statistical Society. Series B (Methodological), **45**, 1 (1983), 47–50.
- [9] S. J. Davey
Extensions to the probabilistic multi-hypothesis tracker for improved data association.
Ph.D. dissertation, The University of Adelaide, Adelaide, South Australia, Sept. 2003.
- [10] S. J. Davey
Simultaneous localization and map building using the probabilistic multi-hypothesis tracker.
IEEE Transactions on Robotics, **23**, 2 (Apr. 2007), 271–280.
- [11] S. J. Davey
Tracking possibly unresolved targets with PMHT.
In *Information, Decision and Control, 2007*, Feb. 2007, 47–52.
- [12] S. J. Davey and D. A. Gray
A comparison of track initiation methods with the PMHT.
In *Information, Decision and Control*, I.E.E.E. Press, Mar. 2002, this paper is found in the printed book; only a conference schedule is on the IEEE's web site.
- [13] S. J. Davey and D. A. Gray
Integrated track maintenance for the PMHT via the hysteresis model.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2007), 93–111.
- [14] S. J. Davey, D. A. Gray, and R. L. Streit
Tracking, association, and classification: A combined PMHT approach.
Digital Signal Processing, **12**, 2–3 (2002), 372–382.
- [15] S. J. Davey, M. G. Rutten, and B. Cheung
A comparison of detection performance for several track-before detect algorithms.
EURASIP Journal on Advances in Signal Processing, **2008** (2008), article ID 428036.
- [16] A. P. Dempster, N. M. Laird, and D. B. Rubin
Maximum likelihood from incomplete data via the EM algorithm.
Journal of the Royal Statistical Society. Series B (Methodological), **39**, 1 (1977), 1–38.
- [17] D. T. Dunham and R. G. Hutchins
Hybrid tracking algorithm using MHT and PMHT.
In *Proceedings of SPIE Conference: Signal and Data Processing of Small Targets*, vol. 4728, Orlando, FL, Apr. 2002, 166–175.
- [18] M. Efe, Y. Ruan, and P. Willett
Probabilistic multi-hypothesis tracker-addressing some basic issues.
In *IEEE Proceedings on Radar, Sonar and Navigation*, vol. 151, no. 4, Aug. 2004, 189–196.
- [19] M. Efe, Y. Ruan, and P. Willett
The pedestrian PMHT.
In *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, Annapolis, MD, 2002, 838–845.
- [20] R. Frühwirth and A. Strandlie
Track fitting with ambiguities and noise: A study of elastic tracking and non-linear filters.
Computer Physics Communications, **120**, 2 (Aug. 1999), 197–214.
- [21] Q. Gan and C. Harris
Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2001), 273–279.
- [22] H. Gauvrit, J. P. Le Cadre, and C. Jauffret
A formulation of multitarget tracking as an incomplete data problem.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1242–1257.
- [23] E. Giannopoulos, R. L. Streit, and P. Swaszek
Multi-target track segment bearings-only association and ranging.
In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, vol. 2, Pacific Grove, CA, Nov. 1997, 1336–1340.
- [24] E. Giannopoulos, R. L. Streit, and P. Swaszek
Probabilistic multi-hypothesis tracking in a multi-sensor multi-target environment.
In *ADFS '96, First Australian Data Fusion Symposium*, Adelaide, SA, Australia, Nov. 1996, 184–189.
- [25] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond
Poisson models for extended target and group tracking.
In *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 5913, San Diego, CA, Aug. 2005, 59 130R-1–59 130R-12.
- [26] C. G. Hempel
The effects of sensor registration error on the performance of PMHT.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, June 2008, 1975–1979.
- [27] C. Hue, J-P. Le Cadre, and P. Pérez
Tracking multiple objects with particle filtering.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (July 2002), 791–812.

- [28] C. Hue, J-P. Le Cadre, and P. Pérez
Sequential monte carlo methods for multiple target tracking and data fusion.
IEEE Transactions on Signal Processing, **50**, 2 (Feb. 2002), 309–325.
- [29] R. G. Hutchins and D. T. Dunhan
Evaluation of a probabilistic multihypothesis tracking algorithm in cluttered environments.
In *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 1996, 1260–1264.
- [30] C. F. Jeff Wu
On the convergence properties of the EM algorithm.
The Annals of Statistics, **11**, 1 (1983), 95–103.
- [31] H. Jeong and J-H. Park
An EM-based adaptive multiple target tracking filter.
International Journal of Adaptive Control and Signal Processing, **16** (2002), 1–23.
- [32] M. Kalandros and L. Y. Pao
Multisensor covariance control strategies for reducing bias effects in interacting target scenarios.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 1 (Jan. 2005), 153–173.
- [33] M. L. Krieg and D. A. Gray
Track fusion in the presence of an interference.
In *Fourth International Symposium on Signal Processing and Its Applications, 1996 (ISSPA '96)*, vol. 1, Aug. 1996, 192–195.
- [34] A. Logothetis, V. Krishnamurthy, and J. Holst
On maneuvering target tracking via the PMHT.
In *Proceedings of the 36th Conference on Decision and Control*, vol. 5, San Diego, CA, Dec. 1997, 5024–5029.
- [35] A. Logothetis, V. Krishnamurthy, and J. Holst
A Bayesian EM algorithm for optimal tracking of a maneuvering target in clutter.
Signal Processing, **82**, 3 (Mar. 2002), 473–490.
- [36] Luginbuhl, P. Ainsleigh, S. Mathews, and R. L. Streit
Accurate likelihood evaluation for multiple model PMHT algorithms.
In *Aerospace Conference 2008*, Mar. 2008.
- [37] T. Luginbuhl and P. Willett
Tracking a general, frequency modulated signal in noise.
In *Proceedings of the 38th Conference on and Control*, Phoenix, AZ, Dec. 1999.
- [38] T. Luginbuhl and P. Willett
Estimating the parameters of general frequency modulated signals.
IEEE Transactions on Signal Processing, **52**, 1 (Jan. 2004), 117–131.
- [39] T. Luginbuhl, Y. Sun, and P. Willett
A track management system for the PMHT algorithm.
In *Proceedings of the 4th International Conference on Information Fusion*, Montreal, Canada, 2001.
- [40] G. J. McLachlan and T. Krishnan
The EM Algorithm and Extensions.
New York: John Wiley & Sons, Inc., 1997.
- [41] K. J. Molnar and J. W. Modestino
Application of the EM algorithm for the multitarget/multi-sensor tracking problem.
IEEE Transactions on Signal Processing, **46**, 1 (Jan. 1998), 115–129.
- [42] T. Moon
The expectation-maximization algorithm.
IEEE Signal Processing Magazine, **13**, 6 (Nov. 1996), 47–60.
- [43] D. Mušicki and X. Wang
Track management and PMHT.
In *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007, 1–5.
- [44] A. G. Pakfiliz and M. Efe
Multi-target tracking in clutter with histogram probabilistic multi-hypothesis tracker.
In *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference*, May 2005, 167–168.
- [45] L. Y. Pao and C. W. Frei
A comparison of parallel and sequential implementations of a multisensor multitarget tracking algorithm.
In *Proceedings of the American Control Conference*, vol. 3, Seattle, WA, June 1995, 1683–1687.
- [46] L. Y. Pao and R. M. Powers
A comparison of several different approaches for target tracking with clutter.
In *Proceedings of the 2003 American Control Conference*, vol. 5, Denver, CO, June 2003, 3919–3924.
- [47] L. Y. Pao and L. Trailović
The optimal order of processing sensor information in sequential multisensor fusion algorithms.
IEEE Transactions on Automatic Control, **45**, 8 (Aug. 2000), 1532–1536.
- [48] M. Petkovšek, H. S. Wilf, and D. Zeilberger
A = B.
Wellesley, MA: A K Peters, Ltd., 1996, ch. 3.
- [49] G. W. Pulford and B. F. La Scala
MAP estimation of target maneuver sequence with the expectation-maximization algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 2 (Apr. 2002), 367–377.
- [50] G. Pulford and A. Logothetis
An expectation-maximisation tracker for multiple observations of a single target in clutter.
In *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, Dec. 1997, 4997–5003.
- [51] C. Rago, P. Willett, and R. L. Streit
Direct data fusion using the PMHT.
In *Proceedings of the American Control Conference*, vol. 3, Seattle, WA, June 1995, 1698–1702.
- [52] C. Rago, P. Willett, and R. L. Streit
A modified PMHT.
In *Proceedings of the 1995 Conference on Information Sciences and Systems*, Mar. 1995.
- [53] C. Rago, P. Willett, and R. Streit
A comparison of the JPDAF and PMHT tracking algorithms.
In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)*, vol. 5, Detroit, MI, May 1995, 3571–3574.
- [54] R. A. Redner and H. F. Walker
Mixture densities, maximum likelihood and the EM algorithm.
SIAM Review, **26**, 2 (Apr. 1984), 195–239.
- [55] A. Roche
EM algorithm and variants: An informal tutorial.
CEA, Tech. Rep., 2003.
- [56] Y. Ruan and P. Willett
The turbo PMHT.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 4 (Oct. 2004), 1388–1398.
- [57] A. Strandlie and J. Zerubia
Particle tracking with iterated Kalman filters and smoothers: the PMHT algorithm.
Computer Physics Communications, **123**, 1 (Dec. 1999), 77–86.
- [58] R. L. Streit
Tracking on intensity-modulated data streams.
NUWC-NPT, Tech. Rep. 11,221, May 2000.

- [59] R. L. Streit
PMHT algorithms for multi-frame assignment.
In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, July 2006.
- [60] R. L. Streit and T. E. Luginbuhl
A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning.
In *Proceedings of the Sixth Joint Service Data Fusion Symposium*, Laurel, Maryland, June 1993, 1015–1024.
- [61] R. L. Streit and T. E. Luginbuhl
Probabilistic multi-hypothesis tracking.
NUWC-NPT, Tech. Rep. 10,428, Feb. 1995.
- [62] N. Ueda and R. Nakano
Deterministic annealing EM algorithm.
Neural Networks, **11**, 2 (Mar. 1998), 271–282.
- [63] M. Verhaegen
Improved understanding of the loss-of-symmetry phenomenon in the conventional Kalman filter.
NASA, Tech. Rep. Technical Memorandum 100041, Dec. 1987.
- [64] M. J. Walsh, M. L. Graham, R. L. Streit, T. Luginbuhl, and L. Mathews
Tracking on intensity-modulated sensor data streams.
In *Proceedings of the IEEE Aerospace Conference*, vol. 4, Big Sky, MT, 2001, 4/1901–4/1909.
- [65] M. J. Walsh
Computing the observed information matrix for dynamic mixture models.
NUWC-NPT, Tech. Rep. 11,768, Sept. 2006.
- [66] M. Wieneke and W. Koch
The PMHT: Solutions for some of its problems.
In *Proceedings of SPIE Conference: Signal and Data Processing of Small Targets Conference*, vol. 6699, San Diego, CA, Aug. 2007, 1–12.
- [67] M. Wieneke and W. Koch
On sequential track extraction within the PMHT framework.
EURASIP Journal on Advances in Signal Processing, **2008** (2008), article ID 276914.
- [68] M. Wieneke and P. Willett
On track-management within the PMHT framework.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, June 2008, 736–743.
- [69] P. Willett, Y. Ruan, and R. Streit
The PMHT for maneuvering targets.
In *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 3373, Orlando, FL, Apr. 1998, 416–427.
- [70] P. Willett, Y. Ruan, and R. L. Streit
A variety of PMHTs.
In *Proceedings of the ISIS/PMHT Workshop*, Paris, France, Nov. 1998.
- [71] P. Willett, Y. Ruan, and R. L. Streit
PMHT: Problems and some solutions.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (July 2002), 738–754.



David F. Crouse (S'05) received the B.S. and M.S. degrees in electrical engineering, in 2005 and 2008, respectively, from the University of Connecticut (UConn), Storrs, and the B.A. degree in German from the University of Connecticut in 2006, for which he spent a year with the Ruprecht-Karls-Universität, Heidelberg, Germany.

He is currently pursuing the Ph.D. degree in electrical engineering at UConn. His research interests lie in the areas of statistical signal processing and tracking.



Marco Guerriero (S'06) was born in Salerno, Italy, on June 18th, 1981. He received his B.A.Sc. and M.Sc. (electrical engineering) from the University of Salerno, Italy in 2002 and 2005 respectively. During his M.Sc. he also spent a semester at Delft University of Technology (TUDELFT), Netherlands, where he took graduate courses in electrical engineering.

He is currently working towards his Ph.D. degree in electrical engineering at the University of Connecticut (UConn), Storrs. In the summer of 2007, he was as a visiting scientist at the NATO Undersea Research Centre (NURC) La Spezia, Italy. His research interests lie in the areas of signal processing, with particular focus on distributed detection and estimation in sensor networks, target tracking and data fusion.



Peter Willett (S'83—M'86—SM'97—F'03) received the B.Sc. degree in engineering science in 1982 from the University of Toronto, Toronto, Canada. He received the M.E. and M.S. degrees in 1983 and 1984, respectively, and the Ph.D. degree in 1986, all in electrical engineering, from Princeton University, Princeton, NJ.

He is a professor at the University of Connecticut, Storrs, where he has worked since 1986. His interests are generally in the areas of detection theory, target tracking, and signal processing.

Dr. Willett is an associate editor for the *IEEE Transactions on Systems, Man, and Cybernetics* and the Editor-in-Chief for the *IEEE Transactions on Aerospace and Electronic Systems*.