

The Set MHT

David F. Crouse, Peter Willett
Department of Electrical and Computer Engineering
University of Connecticut
371 Fairfield Way, U-2157
Storrs, CT 06269, USA
Email: {crouse, willett}@enr.uconn.edu

Lennart Svensson, Daniel Svensson
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Email: lennart.svensson@chalmers.se
daniel.svensson@chalmers.se

Marco Guerriero
Elt Elettronica S.p.A.
Via Tiburtina Valeria Km 13
700 - 00131 Roma, Italy
Email: Marco.Guerriero@elt.it

Abstract—We introduce the Set MHT, a tracking algorithm that maintains multiple hypotheses and produces “smooth” estimates without the track coalescence often associated with Minimum Mean Squared Error (MMSE) estimation or the jitter associated with Maximum Likelihood (ML) estimation. It does this by utilizing Minimum Mean Optimal Subpattern Assignment (MMOSPA) estimation techniques coupled with a theoretically-grounded approach for probabilistically determining the identities of the state estimates. Unlike traditional MHT algorithms, the Set MHT does not “forget” uncertainty in target identities, i.e. display an unjustifiably high confidence level in the target identities, as a result of pruning out competing hypotheses. Rather, it uses merging techniques while avoiding the shortcomings of traditional Gaussian mixture reduction trackers.

Keywords: tracking, MMOSPA, target identity, track coalescence

I. INTRODUCTION

There are a number of properties that a radar or sonar operator might want from a tracking algorithm. Namely, they might want to:

- 1) Know how many targets there are.
- 2) Know the locations of the targets.
- 3) Predict the trajectories of the targets.
- 4) Know what the targets are. This could mean knowing:
 - a) Their identity relative to a specific time designated by the operator.
 - b) Their physical nature according to classification information (e.g. warheads, decoys).
- 5) Have a probabilistic model for the uncertainties in the above estimates.

The above requirements consider what a human operator might want to make tactical decisions. Though an algorithm might need little but a probabilistic model, e.g. Joint Multitarget Probability Density Function (JMPDF) [34] and [22] with classification information from which decisions can be made, final tactical decisions will always be *made by a human*, as he is able to understand and compensate for weaknesses in the model.¹

This work was partially supported by the Office of Naval Research under contracts N00014-09-10613 and N00014-10-10412.

¹As an example, in 1983, a Soviet lieutenant colonel, Stanislav Petrov, chose to ignore computer alerts regarding an American nuclear strike attack and not launch a nuclear counterstrike [19].

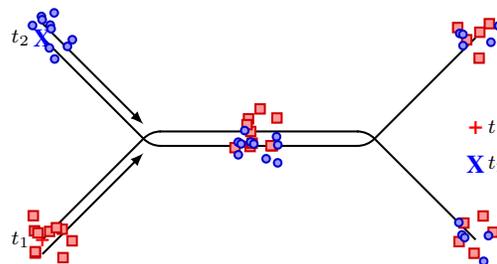


Figure 1: A diagram showing how the mixing of hypotheses between two targets can lead to MMSE estimates (t_1 and t_2) that are between the targets.

It has been observed that the Minimum Mean Squared Error (MMSE) estimate of the target states is “smoother” and more pleasing to an operator compared to the Maximum Likelihood (ML) estimate [9]. On the other hand, it has been noted that the Minimum MSE (MMSE) estimates can cause undesired track coalescence. Indeed, it has been shown that an “optimal” particle filter can be outperformed by an approximation [7], as illustrated in Figure 1, which demonstrates how the MMSE estimates of two targets (the mean) can coalesce when targets are closely-spaced. Though it has been shown that this coalescence problem will resolve itself over time [8], this “resolution” of the problem stems from the from the resampling of the particles, indicating that information in the probability density function (PDF) regarding the uncertainty in the target identities has been lost.² Approaches involving generating hypotheses for estimating the mean that prune “symmetric” hypotheses to avoid coalescence have been considered [24], but the pruning inherently eliminates alternate hypotheses that provide information on the uncertainty of the target identities. On the other hand, the Minimum Mean Optimal Sub-Pattern Assignment (MMOSPA) estimate, discussed in [35] and [17], does not suffer from the problem of coalescence. The Set Joint Probabilistic Data Association Filter (Set JPDAF) [38] is a tracking algorithm that produces approximate MMOSPA estimates [12] [38]. However, the MMOSPA estimate says nothing about the *identities* of the states; that is, how the state estimates are related over time.

²One possible way to lessen the information loss would be to use a particle filter that does not resample the particles, as described in [15]. However, this does not change the fact that MMSE estimates lead to coalescence.

In this paper, we derive a generalization of the Set JPDAF, the Set Multiple Hypothesis Tracker (Set MHT), that utilizes aspects of MMOSPA estimation to produce smooth, coalescence-free state estimates integrated with a framework for estimating the probabilities of the target identities. The Set MHT can also be regarded as a generalization of a Gaussian Mixture Reduction MHT (GMR MHT) [25], [30]. We thus describe a tracker that goes beyond previous algorithms to provide all of the previously-listed information that an operator might desire, with the exception of explicitly estimating the number of targets or using classification information. However, the algorithm can be coupled with initiation and termination methods used in implementations of the MHT and the JPDAF [5], [3], and classification information can be included as has been done elsewhere [16], allowing all of the aforementioned qualities to be satisfied.

In Section II, we consider the foundations of the tracking problem and the accompanying difficulty regarding the explosion in the number of hypotheses. In Section III, we consider the problem of hypothesis reduction and of generating state estimates for display. In Section IV, we look at Gaussian mixture reduction for tracking, and in Section V, we consider how track identities can be maintained when hypotheses are merged both in the Set JPDAF as well as in the Set MHT. The Set MHT is summarized in Section VI and compared to the JPDAF, the Set JPDAF and the GMR MHT in Section VII. The results are concluded in Section VIII.

II. HYPOTHESIS CREATION AND TRACKING

We shall consider the discrete-time stochastic target motion model underlying the basic form of many trackers (see, for example, [2]). The transition of the state, x , for target t from time k to time $k+1$ and any available associated measurements is modeled as

$$x^{k+1}(t) = F^k x^k(t) + v^k(t) \quad z^k(t) = H^k x^k(t) + w^k(t) \quad (1)$$

Here, $z(t)$ is the measurement and $x^k(t)$ represents the state of target t at discrete time k ; $w(t)$ and $v(t)$ are assumed to be independent zero-mean Gaussian white sequences having covariance matrices $R^k(t)$ and $Q^k(t)$.³ Given an initial estimate of the state of a target, if the measurement originating from the target at each time is known, then the optimal recursive estimator of the target's location is the Kalman filter, which is explained in detail in [2] and shall not be repeated here due to space constraints.⁴ In keeping with the notation in [2], we shall let R denote the covariance matrix of the measurement noise, and Q be the covariance of the process noise, which is assumed to be independent of the measurement noise.

In practice, however, the assignment of measurements to tracks is not known and false measurements may be present.

³When target-measurement association uncertainty is a problem, Q and R will be indexed by the measurement and not the target, which is unknown.

⁴The results in this paper can be adapted to scenarios where this model is not adequate. For example, nonlinear target and measurement dynamics may be handled using an Extended Kalman Filter or utilizing Interactive Multiple Model techniques, as described in [2].

In this case, one can build a tree of all possible hypotheses of assigning measurements to tracks, missed detections and clutter (false measurements) over time. Each hypothesis, denoted θ , has a probability of being "correct" given as follows:

$$\Pr \{ \theta | Z^{1:k} \} = \frac{1}{c} \prod_{t=1}^k \prod_{j=1}^{N_T} L_j [\xi_j [\theta^t]] \quad (2)$$

where $Z^{1:k}$ represents all of the measurements up to time k , N_T is the number of targets, c is a constant such that the sum of the probabilities of all of the hypotheses equals one; the function $\xi_j [\theta^t]$ returns the measurement assigned to target j at time t according to hypothesis θ . The value L_j is the likelihood of target j , and, under standard assumptions, letting $P_D^t(j)$ be the probability of detecting target j at time t ; the likelihood is given by Equation (3), where \mathcal{N} represents the normal PDF and λ is the density of the clutter, in points per unit area, which is assumed to be generated uniformly across the observation area with the number of points at each sampling time determined by a Poisson random variable.⁵ This result is also presented in [1] and [3], discussing other formulations under other assumptions.

Given a massive set of hypotheses, we need to determine:

- 1) How to reduce the data in the hypotheses into an estimate of target states that can be presented to the user.
- 2) How to keep the number of possible hypotheses computationally tractable. That is, how to reduce the number of hypotheses at each time step to a tractable number.

We shall discuss both of these issues in the following sections.

III. HYPOTHESIS REDUCTION

A. MMSE Estimation

The JPDAF [3] and the GMR MHT [30], [25] use the MMSE estimate for display. This is given according to [3]

$$\hat{\mathbf{x}} \triangleq \arg \min_{\hat{\mathbf{x}}} \mathbb{E} [\| \mathbf{x} - \hat{\mathbf{x}} \|^2] = \sum_{i=1}^{N_H} w_i \hat{\mathbf{x}}_i \quad (4)$$

where \mathbb{E} denotes the expected value, w_i is the probability of the i th joint association event, of which there are N_H , and $\hat{\mathbf{x}}(i)$ is the stacked set of means of the Gaussian estimates of all N_T targets conditioned on the i th joint association event. The covariance of target t associated with the MMSE estimate is

$$P(t) = \sum_{i=1}^{N_H} w_i \left(P_i(t) + (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}(t)) (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}(t))' \right) \quad (5)$$

where $\hat{\mathbf{x}}(t)$ is the entry in $\hat{\mathbf{x}}$ corresponding to target t and the apostrophe denotes the transpose.

⁵In other words, if n is the number of clutter points at a particular time, then the probability mass function of n is $p(n) = \frac{(\lambda V)^n e^{-\lambda V}}{n!}$, where V is the volume of the observation area.

$$L_j [\zeta_j [\theta^t]] = \begin{cases} 1 - P_D^t(j) & \text{if } \xi_j [\theta^t] = \emptyset \\ \frac{P_D^t(j)}{\lambda} \mathcal{N} [\zeta_j [\theta(t)]; H^t \hat{x}^{t-1}(j), H^t P^{t-1}(j) [H^t]^\prime + R(\xi_j [\theta(t)])] & \text{otherwise} \end{cases} \quad (3)$$

At each time step, the JPDAF approximates all N_H hypotheses as a Gaussian PDF having the mean and covariance given by the MMSE hypothesis.⁶

Though the GMR MHT uses the MMSE estimate for display, it does not use it to reduce the total number of hypotheses. Rather, it clusters the hypotheses into a fixed number of clusters, and then replaces each cluster with its MMSE estimate. Section IV describes how the clustering is performed. Section V describes how target identity can be maintained, both in the Set MHT as well as in the Set JPDAF.

B. MMOSPA Estimation

MMOSPA estimation, first considered in [17], has been discussed in [10], [12], [14], [13], [36] and [35]. We shall summarize some of the more important results. It estimates the target states as

$$\hat{\mathbf{x}}^M \triangleq \arg \min_{\hat{\mathbf{x}}} \mathbb{E} \left[\min_{\mathbf{a}} \|\mathbf{x}(\mathbf{a}) - \hat{\mathbf{x}}\|^2 \right] \quad (6)$$

where \mathbf{a} denotes a particular permutation of the orderings of the states in the vector. This can be evaluated approximately⁷ as

$$\hat{\mathbf{x}}^M \approx \sum_{i=1}^{N_H} w_i \hat{\mathbf{x}}_i(\mathbf{a}_i) \quad (7)$$

where the orderings of the states in each hypothesis are given according to⁸

$$\{\mathbf{a}_1, \dots, \mathbf{a}_{N_H}\} = \arg \max_{\mathbf{a}_1, \dots, \mathbf{a}_{N_H}} (\hat{\mathbf{x}}^M)^\prime (\hat{\mathbf{x}}^M) \quad (8)$$

Appendix A discusses how these ordering can be calculated using quadratic programming. The quadratic programming formulation in this paper has not appeared elsewhere.

It should be noted that because of the reordering of the states in determining the MMOSPA estimate, the relationship between state estimates and their associated tracks is lost. This problem has previously been considered in [37]. In Section V, we show how the state identities can be probabilistically determined. An approximate covariance matrix for the t state estimate is

$$P(t) = \sum_{i=1}^{N_H} w_i \left[P_i(a_i(t)) + [\hat{x}_i(a_i(t)) - \hat{x}(t)] [\hat{x}_i(a_i(t)) - \hat{x}(t)]^\prime \right] \quad (9)$$

The Set JPDAF [38] replaces the set of N_H joint association events at each time-step with a single Gaussian with its mean approximating the MMOSPA estimate and having the

⁶Note that the covariance of $\hat{\mathbf{x}}$ does have cross terms between targets. The JPDAF ignores these. The Coupled JPDAF takes these into account, but it has been shown to not improve the performance of the tracker [3].

⁷This approximation is exact when the hypotheses in question are particles in a particle filter (delta functions) rather than Gaussians, as in the MHT [12].

⁸In some cases, it might be preferable to consider only the position elements of $\hat{\mathbf{x}}^M$ when evaluating (8). This is because the units of the full product make no sense: They are a mixture of position and velocity components.

covariance matrix associated with the approximate MMOSPA estimate of the cluster.

Similar to the relationship between the JPDAF and the GMR MHT, the Set MHT shall use the MMOSPA estimate at each time-step for display, but shall reduce the set of N_H hypotheses to a fixed number, by clustering the hypotheses, reducing each of the clusters to a single Gaussian whose mean and covariance are those associated with the MMOSPA estimate.

IV. CLUSTERING FOR MIXTURE REDUCTION

A. Clustering for the GMR MHT

In [11], a number of techniques for performing Gaussian mixture reduction were considered. A fast approach that performed well is Runnalls' algorithm [28], which successively groups and merges pairs of hypotheses so as to minimize an upper bound on the increase in the Kullback-Leiber (KL) divergence between the original PDF and the reduced PDF. It is given below.

Runnalls' Algorithm

- 1) Set the current mixture to the full mixture.
- 2) The cost of merging components i and j in the current mixture is the upper bound on the increase in the KL divergence [28],

$$c_{i,j} = \frac{1}{2} \left((w_i + w_j) \log [|P_{i,j}|] - w_i \log [|P_i|] - \frac{1}{2} w_j \log [|P_j|] \right) \quad (10)$$

where $P_{i,j}$ corresponds to the full matrix of all N_T targets with block elements determined by (5) for merging only components i and j . Merge the components of the current mixture having the lowest $c_{i,j}$ using (4) and (5) (the weights of the components add, i.e. $w_{12} = w_1 + w_2$) and set the current mixture to the result.

- 3) If the current mixture has the desired number of components, quit. Otherwise, go back to 2.

This is how clustering and mixture reduction can be performed in the GMR MHT. Note that the original version of the GMR MHT used a slightly different approach for reduction, but Runnalls' method has been shown to have better performance [28].

B. Clustering for the Set MHT

For the Set MHT, we shall use a modified version of Runnalls' method that considers pairs of hypotheses in an *unordered* manner. Note that this is a somewhat *ad-hoc* method of determining the clusters, since the upper bound on the KL-divergence behind Runnalls' algorithm no longer holds.

An unordered form of Runnalls' algorithm changes the second step such that $P_{i,j}$ corresponds to the full matrix of all

N_T targets with block elements determined by (9) for merging only components i and j and the components of the current mixture having the lowest $c_{i,j}$ are merged using (7) and (9). In other words, we choose pairs of hypotheses to merge in a somewhat “orderless” manner and the merging operation replaces the hypothesis p_i with a Gaussian whose mean is the MMOSPA estimate having the appropriate, associated covariance. Section V discusses how the relationship between states and tracks in the Set MHT can be determined.

V. MANAGING IDENTITY PROBABILITIES

Unlike the MMSE estimate, the MMOSPA estimate tells us nothing regarding which state estimate corresponds to which track. In deciding how to label our states, we are faced with an ontological⁹ quandary: What constitutes the “best” assignment of states to targets?

In [12], the “best” assignment was considered to either be the one that is the least “jittery” or the one that minimizes a certain approximate probability that a given assignment is closer to the true vector of the track states than any other ordering. In the following subsection, we derive an exact version of the previous probability. In the subsection thereafter, we discuss how this may be computed in a computationally tractable manner. Though everything is discussed in terms of particles, as in a particle filter, we shall use this theory in the Set MHT, whereby individual Gaussians are treated as particles.

A. An Exact Method for Determining Order Probabilities

Suppose that the joint PDF of N_T target states at a particular time is represented by N_H particles. In a typical situation, the i th particle might be represented by p_i ,

$$p_i = \{w_i, \mathbf{x}_i\} \quad (11)$$

where w_i is the probability of the particle, and \mathbf{x}_i is the stacked vector of the target state. Using the approach described in Section III-B, we can use the MMOSPA estimates of the target states to avoid the coalescence problem associated with using the mean (i.e., MMSE) state, as illustrated in (1). However, in order to be able to say something about the identity of the estimates, we will modify the particles.

Let us define a modified form of the particles, \tilde{p}_i , which are completely equivalent to the original particles. The modified particles shall contain an extra component:

$$\tilde{p}_i = \{w_i, \mathbf{x}_i, \mathbf{o}_i\} \quad (12)$$

The addition of the vector \mathbf{o}_i allows us to let each particle represent a family of particles. The vector \mathbf{o}_i contains $N_T!$ elements, the n th one designated as $o_i(n)$. These elements represent the probabilities of the orderings of the states in \mathbf{x}_i conditioned on the fact that this particle represents the “true” set of states of the targets.¹⁰ To convert a normal set

⁹Here, we mean “ontological” in the traditional philosophical sense rather than in the more (annoying) modern computer science sense.

¹⁰In other words, each $o_i(n)$ corresponds to a permutation matrix χ_n . If $o_i(n) = 1$, then that means that if this particle is sampled, the true, ordered states conditioned on this particle being true are $\chi_n \mathbf{x}_i$.

of particles, p_i , into these new particles, one can just insert \mathbf{o}_i into each particle with a one in the slot corresponding to the original ordering of the particles, and a zero elsewhere.

The order vector allows us to change the ordering of the target states in the particles without losing any information; the correct way to reorder each states is encoded in each \mathbf{o} . Suppose that we have a \tilde{p}_i such that the states have been ordered according to the MMOSPA order optimization in (8) and the \mathbf{o} terms have been adjusted accordingly. When we merge all of the particles (sum the w_i ’s and take the means of the other components), we get

$$\mathbf{p}_{\text{merged}} = \{1, \hat{\mathbf{x}}^M, \mathbf{o}^M\} \quad \mathbf{o}^M = \sum_{i=1}^{N_H} w_i \mathbf{o}_i \quad (13)$$

Here, the ordering vector corresponding to the MMOSPA estimate, \mathbf{o}^M , can be interpreted as the probability of the ordering of states in the sample that would be closest to $\hat{\mathbf{x}}^M$ if the underlying distribution were sampled (i.e. we sample the original set of particles). Thus, we have a method of determining the probabilities of the “correct” ordering of the targets in the MMOSPA estimate: The ordering of the tracks that one would most likely want to display to the user is the one with the highest probability in \mathbf{o}^M . To understand how the elements of \mathbf{o}_i should be adjusted when the ordering of the states in \mathbf{x}_i is changed, suppose that we have three tracks and we switch states one and two. The corresponding reordering of the elements in \mathbf{o}_j is shown in Table I.

Thus, given the MMOSPA estimate, $\hat{\mathbf{x}}^M$, of a particle filter from (7) and the orderings of the states in (8), we can determine the probabilities of the orderings of the states in $\hat{\mathbf{x}}^M$ by taking the mean of the correct permutation vector for each particle, \mathbf{o}_i from the ordering information as in (13). When the MMOSPA estimate comes not from a particle filter, but from a set of components in a Gaussian mixture, as was described in Section III-B, then we can approximate the probabilities of the correct ordering by averaging \mathbf{o}_i for each component of the Gaussian mixture, rather than for each particle.

However, the dimensionality of each \mathbf{o}_i is $N_T!$. This is, in general, untenably large. In the following subsection we shall consider a low complexity approximation. For more information on a procedure for maintaining track-to-target probabilities in the SJPDA filter, see [37].

B. Reducing the Complexity of the Order Determination

Each vector \mathbf{o}_i is a Probability Mass Function (PMF) over $N_T!$ permutations. Compressing such PMFs has been considered in numerous works, including [18] and [20].

As noted in [20], if all of the weight in \mathbf{o}_i is on one permutation, then a simple way to reduce the dimensionality of \mathbf{o}_i is to use the permutation matrix corresponding to the ordering described in $\chi_{n(i)}$. Now, if we were to take the expected value of the permutation matrices corresponding to the MMOSPA estimate, instead of the expected value of \mathbf{o}_i as

The Permutation	(1, 2, 3)	(1, 3, 2)	(2, 1, 3)	(2, 3, 1)	(3, 1, 2)	(3, 2, 1)
The Index in \mathbf{o}_j	1	2	3	4	5	6
The Index in $\tilde{\mathbf{o}}_j$	3	4	1	2	6	5

Table I: An example of how the order of the elements of \mathbf{o}_j and $\tilde{\mathbf{o}}_j$ differ if $N_T = 3$ and targets 1 and 2 in \mathbf{x}_j are switched. Each permutation corresponds to an index (and a probability, which is not shown). When two targets are switched, the indices corresponding to each permutation in \mathbf{o}_j must change accordingly as discussed in Section V-A and in [35].

in (13), we would have

$$\chi^M = \sum_{i=1}^{N_H} w_i \chi_i \chi_i^{\text{prev}} \quad (14)$$

where χ_i is the i th permutation matrix corresponding to \mathbf{a}_i in (8) and χ_i^{prev} is the *prior* permutation matrix for the i th particle. In other words, whereas the relationship between the ordering of all $N!$ probabilities had to be shuffled around in a rather complicated manner when the target ordering changes, as in Figure I, we can explicitly express the correct ordering as a product in this simplified example. This method is much simpler, being a $N_T \times N_T$ matrix, thus having only N_T^2 terms rather than $N_T!$ terms for \mathbf{o}^M . Using the example from [20], a possible average permutation matrix might be

$$\chi^M = \begin{bmatrix} & \text{State 1} & \text{State 2} & \text{State 3} \\ \text{Track 1} & 2/3 & 1/6 & 1/6 \\ \text{Track 2} & 1/3 & 1/3 & 1/3 \\ \text{Track 3} & 0 & 1/2 & 1/2 \end{bmatrix} \quad (15)$$

We can see that this gives us marginal probabilities, e.g., we can say that $\Pr\{\text{State 1 is Track 1}\} = 2/3$, but it does not consider joint probabilities. However, we can approximate the joint probabilities as the product of the marginal probabilities, e.g.

$$\Pr\{\text{State 1 is Track 1} \cap \text{State 2 is Track 2} \cap \text{State 3 is Track 3}\} \approx \frac{\chi^M(1,1)\chi^M(2,2)\chi^M(3,3)}{\text{perm}[\chi^M]} \quad (16)$$

where perm represents the matrix permanent (it is the determinant with all $-$ signs in its computation replaced with $+$ signs¹¹). Basically, the denominator comes from the fact that the sum of all possible joint probabilities has to equal one.

Under such an approximation, we can calculate the optimal ordering by choosing exactly one element in each row and column of χ^M so as to maximize the product of their probabilities. This may be solved without enumerating all possible associations using the auction or JVC algorithms [27].

Note that in some instances, the identity matrix might fail to be bistochastic (i.e. all column sums and all row sums equal one) due to precision problems. Methods for making the matrix bistochastic again are discussed in Appendix B.

VI. ALGORITHMIC SUMMARY

The Set MHT is given as follows:

- 1) At each time, given the prior set of hypotheses, the current set of observations is used to build up a posterior (larger) set of hypotheses, as discussed in Section II.

¹¹An efficient method for calculating the matrix permanent is given in [29].

- 2) The hypotheses are reduced using the method of Section IV-B, keeping track of the identities as discussed in Section V.
- 3) The estimate for display is calculated, as discussed in Section III-B.
- 4) One then goes on to the next time-step.

VII. SIMULATIONS

We compared the JPDAF [3], the JPDAF* [6], the Set JPDAF with identity management, the GMR MHT [30], [25] maintaining two hypotheses using Runnalls' method for the clustering, the GMR MHT*¹², and the Set MHT with identity management maintaining two hypotheses in the three scenarios portrayed in Figure 2.

In all scenarios, two targets moved on two-dimensional trajectories from left to right approaching within 50 m of each other and separating. The targets traveled at a constant speed of 170 m/s. All turns performed were done according to the coordinated turn model¹³ described in [2]. In this case, assuming that the two-dimensional state is ordered $\mathbf{x} = [x, \dot{x}, y, \dot{y}]$, the state transition matrix for a sampling interval of T becomes

$$F = \begin{bmatrix} 1 & \frac{\sin[\Omega T]}{\Omega} & 0 & -\frac{1-\cos[\Omega T]}{\Omega} \\ 0 & \cos[\Omega T] & 0 & -\sin[\Omega T] \\ 0 & \frac{1-\cos[\Omega T]}{\Omega} & 1 & \frac{\sin[\Omega T]}{\Omega} \\ 0 & \sin[\Omega T] & 0 & \cos[\Omega T] \end{bmatrix} \quad (17)$$

where Ω is the turn rate in radians per second. We used a turn rate of ± 0.3 rad/s and a constant sampling period of $T = 1/2$ s in all three scenarios. This means that a pilot would feel a force of about 5.3Gs in the turn (one feels one G when standing on the ground), and is a reasonable turn rate for a fighter jet.¹⁴ The angle by which the targets approached and separated was ± 0.6 rad. The middle portion of the flight where the targets were parallel to each other lasted 20 seconds. The straight sections before the first turns and after the final turns were each 10 seconds in duration. The turns were sufficiently long that the targets could achieve and leave parallel flight during the middle segment (2 s). As shown in Figure 2, the three scenarios differed only in the directions of the turns and the starting points of the targets, which were chosen to assure that the targets were 50 m apart during the horizontal, middle section in each scenario.

¹²In the JPDAF* [6], pruning is used to prevent the type of symmetric hypothesis problem that leads to coalescence. The GMR MHT* is the same as the GMR MHT, but with the addition of this pruning when generating joint association events for each predicted hypothesis.

¹³A coordinated turn is one in which speed and altitude do not change.

¹⁴A pilot will typically fall unconscious for turns producing between 4.5 and 5Gs, whereas military pilots wearing 'anti-G' suits can sustain up to 8Gs. Combat aircraft are usually designed for load factors of at least 8Gs [23].

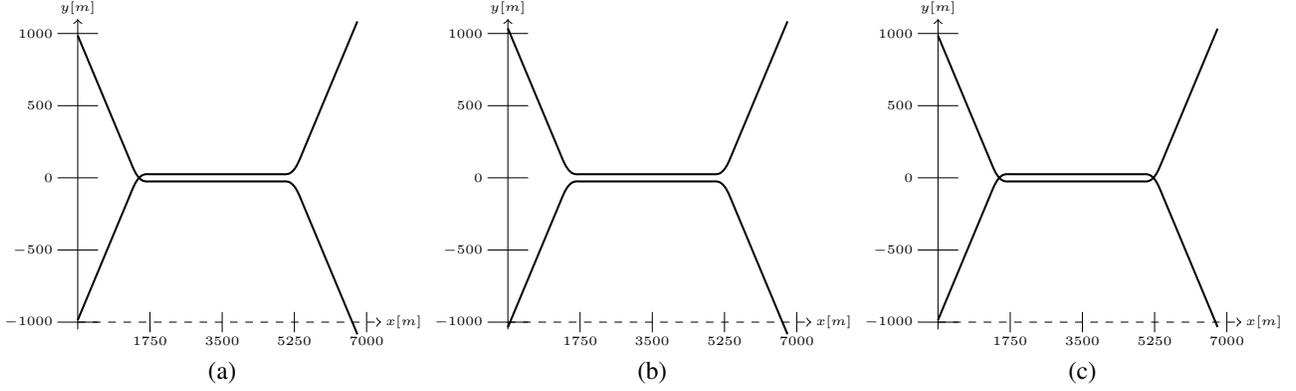


Figure 2: The paths of the tracks in the three scenarios. In (a), the targets cross at the beginning; in (b), the targets never cross and in (c), the targets cross twice. All units are in meters.

Algorithm	Scenario 1		Scenario 2		Scenario 3	
	% Track Loss	P_l	% Track Loss	P_l	% Track Loss	P_l
JPDFAF	6.16	2.00	9.78	47.53	9.14	5.88
JPDFAF*	6.50	1.22	12.88	37.73	11.56	5.30
Set JPDFAF	6.00	2.87	9.62	42.58	8.74	9.63
GMR MHT	7.70	3.77	9.22	50.81	9.88	35.49
GMR MHT*	2.72	8.80	6.04	54.44	3.42	9.41
Set MHT	1.04	1.40	1.08	26.49	0.94	6.65

Table II: The performance of the trackers under each of the scenarios. P_l was calculated using only runs where none of the methods lost tracks. P_l is the % probability of a false labeling of the state estimates at the last time-step.

Two-dimensional measurements were taken in polar coordinates with additive Gaussian noise having standard deviations $\sigma_r = 2$ m and $\sigma_\theta = 1$ mrad and were converted to Cartesian coordinates using the unbiased conversion method of [2]. The measurements from the targets were assumed to always be resolved.¹⁵ A single sensor was placed at (x, y) coordinates of (3.39 km, -30 km). The number of clutter points at each step was determined according to a Poisson process with mean $\lambda = 1 \times 10^{-6} V$ where V is the area of the surveillance region.

Clutter points were placed uniformly over a surveillance region spanning the maximum and minimum x and y positions of the tracks over the whole simulation plus 200 m in each direction. The average number of clutter measurements per scan was approximately 18. Both targets had a detection probability of 80%. Track initialization was done for each track by feeding two correctly associated measurements to an Information Filter [2]. Five thousand Monte Carlo runs were performed for each scenario.

The discretized continuous white-noise acceleration model (DCWNA) [2] was used in the trackers. In accordance with the model, the assumed process noise covariance and state transition matrices for the Kalman filters were

$$Q = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 \\ \frac{1}{2}T^2 & T & 0 & 0 \\ 0 & 0 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ 0 & 0 & \frac{1}{2}T^2 & T \end{bmatrix} q_0 \quad F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

¹⁵In more realistic scenarios, resolution should be taken into account [21]. However, it is peripheral to the issue we are addressing in this paper.

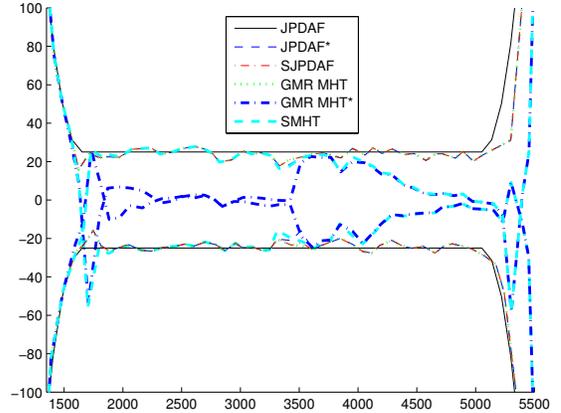


Figure 3: A sample run from Scenario 2 showing how track coalescence makes the GMR MHT perform worse than the JPDFAF and the Set JPDFAF despite propagating more hypotheses, whereas the Set MHT does not suffer this problem.

We used a process noise power spectral density of $q_0 = 400 \text{ m}^2/\text{s}^3$. All of the trackers gated measurements to a 99.97% confidence region, as described in [3].

To evaluate the performance of the trackers in each scenario, we considered the probability of track loss and the probability of a correct labeling of track estimates. The probability of track loss is the probability that one or both of the tracks was lost. Track loss was declared in two ways:

- 1) If the distance from either of the tracks to truth in both possible assignments of tracks to truth exceeded 400 m in either the x or the y direction at the last step of the tracker (a rectangular region), then tracks were declared to have been lost.
- 2) While tracking the targets, joint association events such that the gates for either of the targets exceeded 1,000 m in either position component were dropped. If a tracker lost all of its hypotheses, then the tracks were declared

to have been lost.¹⁶

At each step, one can evaluate the MMOSPA assignment of tracks to truth. If the “labeling” of the targets in the MMOSPA assignment matches the true labeling of the targets, i.e. if we can identify which target was target 1 or target 2 at the first time-step, then the labeling is considered to be correct. In the Set JPDAF and the Set MHT, we used the ordering matrix described in Section V-B and found the most probable labeling of the targets according to the ordering matrix; in the other trackers, the labeling was explicit. If the squared error of the association of tracks to truth was smaller for this labeling than for any other, we declared it to be the “correct” labeling.

Table II shows the results of the simulations. It can be seen that the “set” methods described in this paper with order information have reduced track loss and increased probability of correctly labeling the tracks at the final time-step. The weakness in the GMR MHT is evident in a sample run shown in Figure 3.

VIII. CONCLUSION

We reviewed previous literature on MMOSPA estimation and considered the problem of identity preservation after track merging. We then put these concepts together with an “orderless” Gaussian mixture reduction algorithm to form the Set MHT, a generalization of the Set JPDAF with track identity maintenance, which propagates multiple hypotheses over time, whereas the Set JPDAF propagates only a single hypothesis. It was observed that the Set MHT had lower track loss and maintained the track identities better in all three scenarios than the other trackers maintaining as many or fewer hypotheses. The Set MHT and the Set JPDAF are the only trackers considered that suffer neither track coalescence resulting from MMSE estimation nor express an increasingly false confidence in the target identities after targets separate due to pruning of hypotheses.

REFERENCES

- [1] Y. Bar-Shalom, S. Blackman, and R. J. Fitzgerald, “Dimensionless score function for multiple hypothesis tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 392–400, Jan. 2007.
- [2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, 2001, ch. 11.
- [3] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion*. YBS Publishing, 2011.
- [4] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 2003.
- [5] S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.
- [6] H. A. P. Blom and E. A. Bloem, “Probabilistic data association avoiding track coalescence,” *IEEE Trans. Autom. Control*, vol. 45, no. 2, pp. 247–259, Feb. 2000.
- [7] H. A. P. Blom, E. A. Bloem, Y. Boers, and H. Driessen, “Tracking closely spaced targets: Bayes outperformed by an approximation?” in *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, Jul. 2008.
- [8] Y. Boers, E. Sviestins, and H. Driessen, “Mixed labelling in multitarget particle filtering,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 2, Apr. 2010.
- [9] I. P. Bottlik and S. S. Blackman, “Coordinated presentation of multiple hypotheses in multitarget tracking,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 1096, Sep. 1989, pp. 152–159.
- [10] D. F. Crouse, P. Willett, Y. Bar-Shalom, and L. Svensson, “Aspects of MMOSPA estimation,” in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2011, submitted for Review, March 2011.
- [11] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, “A look at Gaussian mixture reduction algorithms,” in *Proceedings of the 14th International Conference on Information Fusion*, Jul. 2011.
- [12] D. F. Crouse, P. Willett, and Y. Bar-Shalom, “Developing a real-time track display that operators do not hate,” *IEEE Trans. Signal Process.*, accepted March 2010.
- [13] —, “Generalizations of Blom and Bloem’s PDF decomposition for permutation-invariant estimation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2011.
- [14] D. F. Crouse, P. Willett, M. Guerriero, and L. Svensson, “An approximate minimum MOSPA estimator,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2011.
- [15] F. Daum, J. Huang, and A. Noushin, “Exact particle flow for nonlinear filters,” in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XIX*, vol. 7679, 2010.
- [16] O. E. Drummond, “Integration of features and attributes into target tracking,” in *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 4048, Jul. 2000, pp. 610–622.
- [17] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, “Shooting two birds with two bullets: How to find minimum mean OSPA estimates,” in *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, Scotland, Jul. 2010.
- [18] L. J. Guibas, “The identity management problem – a short survey,” in *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, Jul. 2008.
- [19] D. Hoffman, “I had a funny feeling in my gut,” *Washington Post*, Feb. 10, 1999. [Online]. Available: <http://www.washingtonpost.com/wp-srv/inatl/longterm/coldwar/shatter021099b.htm>
- [20] J. Huang, C. Guestrin, and L. J. Guibas, “Fourier theoretic probabilistic inference over permutations,” *Journal of Machine Learning Research*, vol. 10, pp. 997–1070, Jan. 2009.
- [21] W. Koch, “On Bayesian tracking and data fusion: A tutorial introduction with examples,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 29–51, Jul. 2010.
- [22] C. Kreucher, K. Kastella, and A. O. Hero III, “Multitarget tracking using the joint multitarget probability density,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.
- [23] W. A. Mair and D. L. Birdsall, *Aircraft Performance*, ser. Cambridge Aerospace Series 5. Cambridge, UK: Cambridge University Press, 1992.
- [24] R. D. Palkki, A. D. Lanterman, and W. D. Blair, “Addressing track hypothesis coalescence in sequential k -best multiple hypothesis tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, Apr. 2011.
- [25] L. Y. Pao, “Multisensor mixture reduction algorithms for tracking,” *Journal of Guidance, Control and Dynamics*, vol. 17, no. 6, pp. 1205–1211, Nov.–Dec. 1994.
- [26] B. N. Parlett and T. L. Landis, “Methods for scaling to doubly stochastic form,” *Linear Algebra and Its Applications*, vol. 48, pp. 53–79, Dec. 1982.
- [27] K. R. Pattipati, R. L. Popp, and T. Kirubarajan, “Survey of assignment techniques for multitarget tracking,” in *Multitarget-Multisensor Tracking: Applications and Advances*, Y. Bar-Shalom and W. D. Blair, Eds. Boston: Artech House, 2000, vol. III, ch. 2.
- [28] A. R. Runnalls, “Kullback-Leibler approach to Gaussian mixture reduction,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, Jul. 2007.
- [29] H. J. Ryser, *Combinatorial Mathematics*, ser. The Carus Mathematical Monographs. The Mathematical Association of America, 1963, no. 14, ch. 2.
- [30] D. J. Salmond, “Mixture reduction algorithms for point and extended object tracking in clutter,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 2, pp. 667–686, Apr. 2009.

¹⁶In practice, one would generally only drop the target in the joint association event that caused the problem, requiring estimation based on partial hypotheses, which we avoided for simplicity’s sake. Nonetheless, the comparison performed here fair, since the same method of hypothesis rejection was used in all of the trackers. This is a different criterion than used in [35].

- [31] M. H. Schneider and S. A. Zenios, "A comparative study of algorithms for matrix balancing," *Operations Research*, vol. 38, no. 3, pp. 439–455, May–Jun. 1990.
- [32] R. Sinkhorn, "A relationship between arbitrarily positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 876–879, Jun. 1964.
- [33] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.
- [34] L. D. Stone, C. A. Barlow, and T. L. Corwin, *Bayesian Multiple Target Tracking*. Boston: Artech House, 1999, ch. 5.
- [35] D. Svensson, L. Svensson, M. Guerriero, and P. Willett, "On the calculation of minimum mean OSPA estimates," Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, Tech. Rep. R004/2011, 2011.
- [36] D. Svensson, "Target tracking in complex scenarios," Ph.D. dissertation, Chalmers University of Technology, Göteborg, Sweden, 2010, pp. 111–146.
- [37] D. Svensson, L. Svensson, M. Guerriero, D. F. Crouse, and P. Willett, "The multitarget Set JPDA filter with target identity," in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target Recognition XX*, 2011.
- [38] L. Svensson, D. Svensson, and P. Willett, "Set JPDA algorithm for tracking unordered sets of targets," in *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, Jul. 2009, pp. 1187–1194.

APPENDIX A

FORMULATING THE MMOSPA OPTIMIZATION

In [36], an algorithm for determining the optimal ordering of the hypotheses in the MMOSPA estimate was derived as a continuous quadratic programming problem over $N_T!N_H$ variables. Here we derive a similar, optimal algorithm that requires only $N_T^2(N_H - 1)$ variables.

We shall express the components of the MMOSPA estimate as

$$\hat{\mathbf{x}}^M = [\hat{x}^M(1)', \hat{x}^M(2)', \dots, \hat{x}^M(N_T)']' \quad (19)$$

where $\hat{x}^M(i)$ is the i th state estimate.

Fixing the ordering of the first hypothesis to reduce ambiguity in the solution, we can thus rewrite (7) to express the MMOSPA state estimate for track t as

$$\hat{x}^M(t) = w_1 \hat{x}_1(t) + \sum_{i=2}^{N_H} \sum_{j=1}^{N_T} w_i \phi_i^t(j) \hat{x}_i(j) \quad (20)$$

$$= w_1 \hat{x}_1(t) + V(t) \phi^t \quad (21)$$

where $\phi_i^t(j) \in \{0, 1\}$ and $\phi_i^t(j) = 1$ means that state j goes to track t in hypothesis i in the mixture and

$$\phi(t) = [\phi_2^t(1), \phi_2^t(2), \dots, \phi_2^t(N_T), \phi_3^t(1), \dots, \phi_{N_H}^t(N_T)]' \quad (22)$$

$$V(t) = [w_2 \hat{x}_2(1), w_2 \hat{x}_2(2), \dots, w_2 \hat{x}_2(N_T), w_3 \hat{x}_3(1), \dots, w_{N_H} \hat{x}_{N_H}(N_T)] \quad (23)$$

$V(t)$ is $D \times N_T(N_H - 1)$ and $\phi(t)$ is $N_T(N_H - 1) \times 1$.¹⁷

Thus, we can write

$$\hat{\mathbf{x}}^M = w_1 \hat{\mathbf{x}}_1 + \mathbf{V} \phi \quad (24)$$

¹⁷Note that N_H unto itself can be quite large. Given N_T targets with no clutter and no missed detections, whereby all measurements gate with all targets, $N_H = N_T!$

where

$$\mathbf{V} = \text{diag}[V(1), \dots, V(N_T)] \quad \phi = [\phi(1), \dots, \phi(N_T)]' \quad (25)$$

and ϕ is $N_T^2(N_H - 1) \times 1$. Note that $V(1) = V(2) \dots = V(N_T)$.

The optimization problem consists of determining ϕ . After dropping constant terms, the optimization of (8) becomes

$$\begin{aligned} & \max_{\phi} \frac{1}{2} \phi' \mathbf{V}' \mathbf{V} \phi + w_1 \hat{\mathbf{x}}_1' \mathbf{V} \phi \\ & \text{subject to } \sum_j \phi_i^t(j) = 1 \quad \forall \{i, t\} \quad \phi_i^t(j) \geq 0 \quad \forall \{i, j, t\} \\ & \sum_t \phi_i^t(j) = 1 \quad \forall \{i, j\} \end{aligned}$$

This represents $2N_T(N_H - 1)$ equality constraints, some of which are dependent, and $N_T^2(N_H - 1)$ inequality constraints.

Note that $\mathbf{V}' \mathbf{V}$ is positive semidefinite, meaning that the optimal point will be along the border region of the constraints. In other words, the optimal solution will be such that $\phi_i^t(j) \in \{0, 1\}$, thus eliminating the need to explicitly constrain the solution. This is a (degenerate) quadratic optimization problem and can be solved (approximately) using various active set techniques, as described in [4], or with certain optimization packages, such as `quadprog` in MATLAB.

APPENDIX B

ALGORITHMS FOR BISTOCHASTIC NORMALIZATION

Over time, due to numeric errors in the mixing as a result of the finite precision in digital computer systems, the identity-probability matrices described in Section V-B can become no longer bistochastic; that is, the row sums and the column sums of the matrix are no longer all equal to one. In such an instance, we would like to bistochastically normalize the matrices, altering the individual elements as little as possible.

The problem of bistochastic normalization has been previously studied in [26] and [31], among other sources. The simplest such algorithm, often called Sinkhorn's algorithm and is discussed in [32] and [33].

We suggest an alternative approach; we would like to modify the matrix O as little as possible and still make it bistochastic. To do that, we will find a bistochastic matrix X with all positive elements, that differs as little as possible from O in terms of the Frobenius norm, which is defined to be

$$\|O\|_F \triangleq \sqrt{\text{Tr}[OO']} \quad (26)$$

The solution for X turns out to be a quadratic programming problem:

$$\begin{aligned} & \min_{x_{i,j}} \sum_{i=1}^D \sum_{j=1}^D (x_{i,j} - o_{i,j})^2 \\ & \text{subject to } \sum_{i=1}^D x_{i,j} = 1 \quad \forall j \quad \sum_{j=1}^D x_{i,j} = 1 \quad \forall i \\ & x_{i,j} \geq 0 \quad \forall \{i, j\} \end{aligned}$$

and can be solved using using common optimization packages, such as `quadprog` in MATLAB.