JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 3, May–June 2007

Multiple-Target Tracking and Identity Management with Application to Aircraft Tracking

Inseok Hwang*
Purdue University, West Lafayette, Indiana 47907
and
Hamsa Balakrishnan,† Kaushik Roy,‡ and Claire Tomlin§
Stanford University, Stanford, California 94305

DOI: 10.2514/1.27366

The problem of tracking and managing the identities of multiple targets is discussed and applied to the passive radar tracking of aircraft. The targets are assumed to be commercial aircraft switching modes during flight, and are thus well modeled by hybrid systems. We propose a computationally efficient algorithm based on joint probabilistic data association for target-measurement correlation. We use the results of this algorithm to simultaneously implement an identity management algorithm based on identity-mass flow, and a multiple-target tracking algorithm based on the residual-mean interacting multiple model algorithm. Together, they constitute the multiple-target tracking and identity management algorithm. The multiple-target tracking and identity management algorithm incorporates suitable local information about target identity, when available, in a manner that decreases the uncertainty in the system as measured by its statistical entropy. For situations in which local information is not explicitly available, a technique based on multiple hypothesis testing is proposed to infer such information. This algorithm allows us to track multiple targets, each capable of multiple modes of operation, in the presence of continuous process noise and of spurious measurements. The multiple-target tracking and identity management algorithm is demonstrated through various scenarios that are motivated by air traffic surveillance applications.

I. Introduction

THE multiple-target tracking problem deals with correctly tracking several targets using a collection of noisy sensor measurements that are collected at each instant in time. The identity management problem tries to associate target identities with these tracks, or state estimates. Although closely related, the two problems have traditionally been studied independently. In this paper, we propose an algorithmic framework for multiple-target tracking and identity management, which is, to our best knowledge, the first attempt to combine the two problems in a systematic way.

The current air traffic surveillance system uses data from radar measurements to track aircraft. In spite of a substantial improvement in technology, the radar system is still vulnerable to several challenges, such as the large number of aircraft, extraneous measurements from clouds, birds, and other objects, as well as "phantom" blips [1]. Another issue that poses a challenge is the growing number of general aviation aircraft. These aircraft do not transmit their identities unless their transponders are switched on, and even then, the transponders are fraught with problems [2]. Because air traffic controllers are instructed not to issue orders to aircraft unless they are certain of their identity [3], it becomes essential that they have access to reliable track data with the associated aircraft identities, so that they can maintain safety in the flow of aircraft. The simultaneous execution of both tasks would be useful for air traffic surveillance, because air traffic control advisories are based on estimates of the aircraft situation data, which consist of the state estimates and identity estimates for all the aircraft in the relevant region of airspace. In the current air traffic control system, if a controller detects a conflict but does not know the identity of the aircraft involved in the conflict, he or she tries to identify the aircraft through voice communications [4]; there is no automated algorithm available for simultaneous aircraft tracking and identity management. Although the chief emphasis of this work is in the development of an advisory tool for air traffic controllers, the theory we develop is applicable to tracking problems in more general sensor networks [5–8]. The applications of the proposed algorithm include land, sea, air, and space surveillance for military use; and collision avoidance, navigation, and image processing for civilian use.

In practice, the target tracking and identity management problem is complicated by the limitations in the quality of available information about the targets, as well as the presence of signals, known as clutter. The behavior of the targets also adds complexity to the problem: target interactions increase the uncertainty in the system. These issues motivate the development of a combined target tracking and identity management algorithm that can be deployed in cluttered environments.

For multiple-target tracking in clutter, we have to decide which measurement is associated to which target and which measurements are clutter. This problem has been addressed by several data association algorithms, which associate measurement data with targets [1,9]. One such algorithm is the joint probabilistic data association (JPDA) algorithm in which target kinematic information (position and velocity) is used for associating measurements with targets. However, the JPDA algorithm is computationally expensive, and might be impractical when tracking many aircraft in a cluttered environment [10]. For this reason, several modified versions of the JPDA algorithm have been proposed [10–12]. The approximate JPDA algorithm proposed in [10] is useful for multiple-target tracking, but unfortunately may not result in a marginal probability distribution of measurement-target associations, which represents

Received 18 August 2006; revision received 28 November 2006; accepted for publication 8 December 2006. Copyright © 2007 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/07 $10.00 in correspondence with the CCC.

*Assistant Professor, School of Aeronautics and Astronautics; ihwang@purdue.edu. Member AIAA.

†University of California at Santa Cruz, Moffett Field, CA 94035; hamsa@stanfordalumni.org. Member AIAA.

‡Electrical Engineering Department; kroy1@stanford.edu. Student Member AIAA.

§Associate Professor, Department of Aeronautics and Astronautics; and Associate Professor, Department of Electrical Engineering and Computer Science, University of California, Berkeley; tomlin@stanford.edu. Member AIAA.

the physical constraint that there is only one measurement from one target. We propose a modified version of the approximate JPDA [10], which satisfies this physical constraint.

Assignment algorithms have also been used to overcome the computational complexity of data association in multiple-target tracking problems [13,14]. These algorithms minimize the sum of distances between measurements and the expected target positions. This implies that, like the nearest-neighbor data association algorithm, these algorithms select the measurement that is closest to the predicted measurement, without considering all possible measurement-target correspondences. Therefore, they lose some of the advantages of the JPDA algorithm, which considers all possible correlations between measurements and targets. As an alternative, we propose the use of data association probabilities, computed by our modification of the approximate JPDA algorithm, as weighting coefficients for the assignment problem. We then use the extended Munkres algorithm [15,16] to maximize the overall data association probability. In this way, we can consider measurement-target correlation like JPDA.

Most currently available multiple-target data association and tracking algorithms emphasize the need to track several targets simultaneously, yet they do not address the need to distinguish between the different targets, and indeed, often lead to target-swapping while tracking. In this paper we propose an identity management algorithm, which can use local attribute information about targets to keep track of the identities of the targets. In some situations in which local identity information from sensors is not readily available, we obtain this information using the multiple hypothesis testing (MHT) algorithm [17,18] to discern the identities of the targets when the targets are close to each other, and therefore their identities are mixed. By logically combining these algorithms, we develop the multiple-target tracking and identity management (MTIM) algorithm, which can keep track of multiple aircraft and their identities.

The rest of this paper is organized as follows: In Sec. II, we introduce the multiple-target tracking and identity management problem, and discuss the basic components of the algorithmic framework that we propose to solve it. Section III describes the multiple-target tracking and identity management algorithm for a cluttered environment. We describe, in Sec. IV, the use of the algorithm for a special case in which there is no clutter. Simulations for multiple-aircraft tracking scenarios are presented as demonstration of the ability of the proposed algorithm for simultaneous multiple-target tracking and identity management to perform in cluttered and noncluttered environments. Finally, conclusions are presented in Sec. V.

II. Algorithmic Framework

In this section, we consider the problem of associating a time series of measurements to the tracks of multiple targets, and managing their identities. The MTIM algorithm approaches this problem using the structure shown in Fig. 1 at each time step. Throughout the paper, we assume that the number of targets is known and constant, though in recent work we have developed algorithms that can deal with the unknown and time-varying numbers of targets [8,19,20]. The algorithm is broken up into three stages as shown in Fig. 1.

The first stage is "measurement validation and data association," which consists of matching incoming measurements to the targets. Suppose there are N targets. Given state predictions and covariances of the N targets and L ($\geq N$) measurements from the current time step, the data association block is used to generate an $L \times N$ matrix of association probabilities. Note that for noncluttered environments, $L = N$, and for clutter, $L \geq N$. Entries in this matrix represent the probability of a given measurement (indexed by the row) having originated from a given target (indexed by the column). Because there are N targets, we should select N measurements from L measurements because only one measurement can originate from a target. In scenarios with no clutter and no undetected targets, this is not a problem because the number of measurements equals the number of targets, that is, the data association matrix is square.

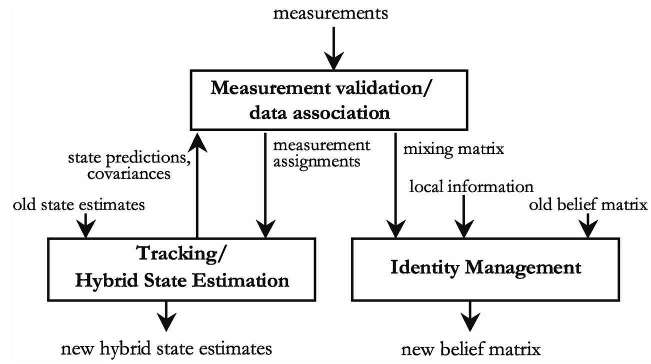


Fig. 1 Structure of the MTIM algorithm for one time step.

Finally, the data association block uses the state predictions for each of the targets, obtained from the hybrid state estimation block, to associate a single measurement with each of the targets (the measurement assignments).

The "tracking or hybrid state estimation" block executes the tracking of N targets in parallel. The tracking algorithm for each target takes as input the hybrid state estimate from the previous time step and a single measurement for the current time. The hybrid state estimates from the previous time step are sufficient to compute the state predictions for the N targets, which are passed on to the measurement validation/data association block. The measurement input comes from the data association block. For air traffic surveillance scenarios, the hybrid state estimate is composed of position and velocity estimates, their covariances, and a flight mode estimate. The final output of the tracking/hybrid state estimation block is the hybrid state estimates for the current time step.

The "identity management" block takes as input the belief matrix from the previous time step and the $N \times N$ mixing matrix output by the data association block. Entries in the belief matrix represent the probability that a given target has a given identity. The belief matrix is initialized to the identity matrix, because it is assumed that all targets are initially uniquely identified. The mixing matrix stores interaction information for a single time step. The ijth element of this matrix represents the probability that target i at the previous time step has become target j at the current time step. The identity management block outputs the belief matrix for the current time step.

The following sections discuss each block and the algorithms used to implement the different stages in detail.

A. Measurement Validation and Data Association Block

We denote $z(k)$ as a measurement of target position observed at time k, and $\hat{z}(k|k-1)$ as a predicted measurement at time k using the (continuous) measurement information up to time $k-1$.

Definition 1: The validation gate at time k, which we denote $\tilde{V}_k$, is defined as

$$\tilde{V}_k(\gamma) := \{z(k) | r(k)^T S^{-1}(k) r(k) \leq \gamma^2\} \quad (1)$$

where $r(k) = z(k) - \hat{z}(k|k-1)$ is the residual, $S(k)$ is its covariance, and γ is a design parameter that determines the size of the validation gate.

Measurements that lie inside the gate at each time step are considered validated. The set of validated measurements at time k is denoted by

$$Z(k) := \{z_i(k)\}_{i=1}^{m_k} \quad (2)$$

where m_k is a random variable that represents the number of validated measurements at that time step. The validated measurement sequence up to time k is defined as

$$Z^k := \{Z(j)\}_{j=1}^{k} \quad (3)$$

We make the standard assumption that

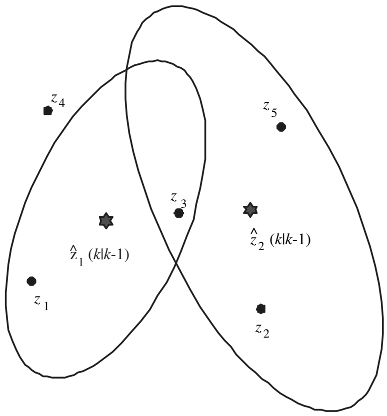


Fig. 2 Validation gates at time *k*.

$$p[z(k)|Z^{k-1}] = \mathcal{N}[z(k); \hat{z}(k|k-1), S(k)]$$

where $\mathcal{N}[x; a, b]$ denotes a Gaussian random variable x with mean a and variance b.

This problem of either associating every validated measurement with an appropriate target, or identifying it as clutter and discarding it, is known as data association. We note that the set of validated measurements could potentially consist of both correct and incorrect measurements. For example, Fig. 2 shows that measurements z_1 and z_3 are validated for target 1 (with predicted measurement $\hat{z}_1$), measurements z_2, z_3, and z_5 are validated for target 2 (with predicted measurement $\hat{z}_2$), and z_4 is not validated for either target at time k. For data association, we consider the JPDA algorithm [21–23] in this paper.

The JPDA algorithm is an extension of the probabilistic data association (PDA) algorithm, used for tracking a single target [1,18] in a cluttered environment. The JPDA algorithm was proposed for associating measurements to multiple targets in the presence of random noise, or clutter. It is a suboptimal approximation of optimal (Bayesian) filtering, and consists of sequentially associating current tracks with the most recent observations. The association probabilities are computed for all possible track-measurement associations. The JPDA algorithm can be shown to be more effective than nearest-neighbor association, in which the single best association is selected for each target, especially in cluttered environments [1].

In the PDA algorithm, only one of the (possibly many) validated measurements is assumed to have originated from the target being tracked. The other measurements are assumed to be clutter (that is, false alarms) and are modeled as independent identically distributed (i.i.d) random variables with uniform spatial distributions. The basic assumption of the PDA algorithm is that

$$p[x(k)|Z^{k-1}] = \mathcal{N}[x(k); \hat{x}(k|k-1), P(k|k-1)] \tag{4}$$

that is, the continuous state is assumed to be distributed as a Gaussian random variable, whose mean and covariance are given by the latest predicted state $\hat{x}(k|k-1)$ and its covariance $P(k|k-1)$, respectively.

The JPDA algorithm is used when the number of targets in a cluttered environment is known a priori. If there are several targets in the same neighborhood, measurements from one target can consistently fall inside the validation gates of neighboring targets, degrading the performance of the PDA algorithm. This phenomenon can result in track swapping, where the tracks of the two targets are swapped, and track coalescence. The key to the JPDA algorithm is the evaluation of the conditional probabilities of joint events: measurement j originated from target t, that is, a consistent association of every target to a measurement. For example, a possible joint event Θ for the example shown in Fig. 2 is that measurement z_3 corresponds to target 1, measurement z_2 corresponds to target 2, and measurements z_1, z_4, and z_5 are clutter.

The joint event association matrix can be represented by the permutation matrix, $\Omega = [\omega_{jt}(\Theta)]$, where Ω is a $m_k \times (N+1)$ matrix. $\omega_{jt} = 1$ for an event in which measurement j originates from target t. The first column in Ω corresponds to events in which the measurement j does not correspond to any target, that is, $\omega_{j0} = 1$ for an event in which measurement j is clutter. The clutter is assumed to be uniformly distributed in the surveillance region of volume V. We define the following notation:

$$\begin{aligned} &\delta_t(\Theta) = \sum_{j=1}^{m_k} \omega_{jt}(\Theta) \le 1\text{: target detection indicator} \\ &\tau_j(\Theta) = \sum_{t=1}^{N} \omega_{jt}(\Theta)\text{: measurement association indicator} \\ &\phi(\Theta) = \sum_{j=1}^{m_k} [1 - \tau_j(\Theta)]\text{:} \\ &\qquad \text{number of unassociated measurements in} \Theta \\ &\qquad P_D\text{: target detection probability of target } t \end{aligned} \tag{5}$$

Then, the marginal association probability, that is, the probability that measurement j belongs to target t, is given by

$$\begin{aligned} \beta_{jt} &= \sum_{\Theta} p\{\Theta|Z^k\}\omega_{jt}(\Theta), \quad \text{where } p\{\Theta|Z^k\} \\ &= \frac{1}{c}\frac{\phi!}{V^\phi} \prod_{j=1}^{m_k} \{\mathcal{N}_{t_j}[z_j(k)]\}^{\tau_j} \prod_{t=1}^{N} (P_D^t)^{\delta_t}(1 - P_D^t)^{1-\delta_t} \\ \mathcal{N}_{t_j}[z_j(k)] &= \mathcal{N}[z_j(k); \hat{z}^{t_j}(k|k-1), S^{t_j}(k)] \end{aligned} \tag{6}$$

In the preceding expressions, $\hat{z}^{t_j}(k|k-1)$ denotes the predicted measurement for target t_j with an associated covariance S^{t_j} at time k. We note here that summing over all possible joint events becomes computationally intractable as the number of aircraft increases, because the number of joint events is $\frac{m_k!}{(m_k-N-1)!}$. A more thorough introduction to data association algorithms can be found in [24]. A more detailed explanation of PDA and JPDA is provided in [21].

B. Hybrid State Estimation Block

Once a measurement is associated to each target, we use a hybrid estimation algorithm to compute the state estimates of each target while tracking multiple maneuvering targets as shown in Fig. 1. This component of the algorithm is therefore called the tracking/hybrid state estimation block. In this section, we briefly review the general structure of the interacting multiple model (IMM) algorithm [17,24], and then a modification of the IMM algorithm [25,26], which we had proposed.

In this paper, targets are assumed to be commercial aircraft switching modes during flight and thus we model the dynamics of an aircraft as a stochastic linear hybrid system [27] with discrete-time, continuous-state dynamics given by

$$x(k+1) = A_j x(k) + w_j(k) \qquad z(k) = C_j x(k) + v_j(k) \tag{7}$$

and a Markov transition of the discrete state (also known as the mode) given by

$$p[j(k+1)|i(k)] = \Phi_{ij}, \qquad i, j \in \{1, 2, \cdots, N_{\text{modes}}\} \tag{8}$$

where $x(k) \in \mathbb{R}^n$ and $z(k) \in \mathbb{R}^p$ are the continuzous-state variable and the (continuous) output, respectively, at time k. We assume that the number of modes is N_{modes}. The terms w_j and v_j are the mode-dependent, uncorrelated, white Gaussian process noise and measurement noise, with zero means, and covariances Q_j and R_j, respectively. Φ_{ij} is the (Markovian) mode transition probability from mode i to mode j, and is assumed to be constant. An advantage of using a hybrid system as the dynamics of an aircraft is that simple kinematic models can be used in individual modes because in each mode, the aircraft dynamics is simple (i.e., straight flight or

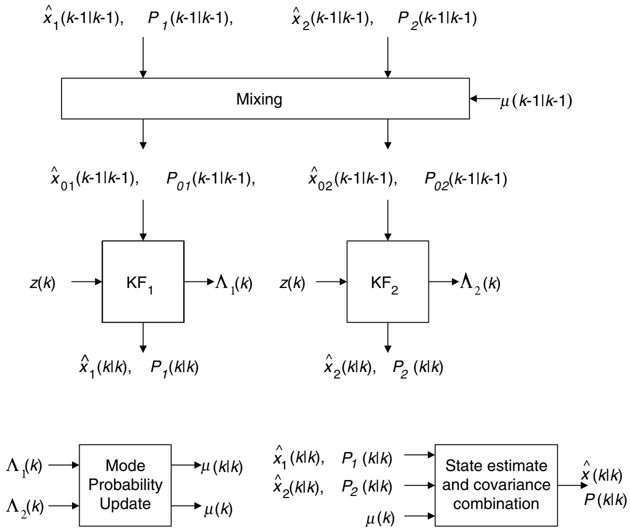


Fig. 3 Structure of the IMM algorithm (for two modes) [33].

coordinate turn). Without using a complex dynamic model of an aircraft for tracking, which requires accurate system parameters that may not be known to surveillance systems, we can improve performance of aircraft tracking [1,25,26].

Given the system parameters of the model in (7) and (8), hybrid state estimation requires the estimation of both the continuous state and the discrete state at time k, using the measurement sequence up to time k. The IMM algorithm is a multiple-model-based state estimation algorithm that computes the hybrid state estimate at every time step using a weighted sum of estimates from a bank of Kalman filters, each matched to the dynamics of a different mode of the system. Figure 3 shows the general structure of the IMM algorithm, for a stochastic linear hybrid system with two modes. A description of the different components of the IMM algorithm is as follows:

1) Mixing probability: This is the probability that the system was in mode i at time $k-1$, given the continuous output of the system up to time $k-1$, and given that it is in mode j at time k, that is,

$$\mu_{ij}(k-1|k-1) = \frac{1}{c_j}\Phi_{ij}\mu_i(k-1) \tag{9}$$

where c_j is a normalization constant, and $\mu_i(k-1)$ is the mode probability of mode i at time $k-1$, that is, a measure of how probable it is that the system (the target being tracked) is in mode i at time k. The initial condition $\mu_i(0)$ is assumed given, and is usually obtained from properties of the system.

2) New initial states and covariances: The input to each Kalman filter (KF_i) is adjusted by weighting the output of each Kalman filter with the mixing probability as the weight:

$$\hat{x}_{0j}(k-1|k-1) = \sum_{i=1}^{N_{\text{modes}}} \hat{x}_i(k-1|k-1)\mu_{ij}(k-1|k-1)$$
$$P_{0j}(k-1|k-1) = \sum_{i=1}^{N_{\text{modes}}} \{P_i(k-1|k-1) + [\hat{x}_i(k-1|k-1) - \hat{x}_{0j}(k-1|k-1)][\hat{x}_i(k-1|k-1) - \hat{x}_{0j}(k-1|k-1)]^T\}\mu_{ij}(k-1|k-1)$$

where $\hat{x}_i(k-1|k-1)$ and $P_i(k-1|k-1)$ are the state estimate and its covariance produced by Kalman filter i at time $k-1$, after the measurement update step.

3) Kalman filter: N_{modes} Kalman filters run in parallel [multiple-model-based (hybrid) estimation]. The outputs of Kalman filter matched to mode j (KF_j) are the continuous-state estimate $\hat{x}_j(k|k)$, its covariance $\hat{P}_j(k|k)$, and the mode likelihood function $\Lambda_j(k)$ described next.

4) Mode likelihood functions: The likelihood function of mode j is a measure of how likely it is that the model used in Kalman filter j is the correct one; it is computed with the residual and its covariance produced by Kalman filter j:

$$\Lambda_j(k) = \mathcal{N}[r_j(k); 0, S_j(k)] \tag{10}$$

where $r_j(k) := z(k) - C_j\hat{x}_j(k|k-1)$ is the residual of Kalman filter j, $\hat{x}_j(k|k-1)$ is a state estimate by Kalman filter j at time k before the measurement update, and $S_j(k)$ is its covariance.

5) Mode probabilities: The probability of mode j is a measure of how probable it is that the system is in mode j:

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\sum_{i=1}^{N_{\text{modes}}}\Phi_{ij}\mu_i(k-1) \tag{11}$$

where c is a normalization constant. The probability of each mode is updated using the likelihood function.

6) Output: The outputs of IMM are the continuous-state estimate $\hat{x}(k|k)$, its covariance $P(k|k)$, and the mode estimate $\hat{m}(k|k)$:

$$\hat{x}(k|k) = \sum_{j=1}^{N_{\text{modes}}} \hat{x}_j(k|k)\mu_j(k)$$
$$P(k|k) = \sum_{j=1}^{N_{\text{modes}}} \{P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)][\hat{x}_j(k|k) - \hat{x}(k|k)]^T\}\mu_j(k) \tag{12}$$
$$\hat{m}(k|k) = \arg\max_j \mu_j(k)$$

The continuous-state estimate is a weighted sum of the estimates from N_{modes} Kalman filters and the mode estimate is the mode which has the highest mode probability.

In this paper, for hybrid state estimation, we use the residual-mean interacting multiple model (RMIMM) algorithm. RMIMM uses a new likelihood function by using the mean of the residuals to achieve improved hybrid state estimation performance, especially fast mode transition estimation. The new likelihood function is given by [25,26]

$$\Lambda_j^{\text{new}}(k) = \begin{cases} \frac{N_j(k)\Lambda_j(k)}{\sum_{i=1}^{N_{\text{modes}}} N_i(k)\Lambda_i(k)} & \text{if } \bar{r}_j(k) \neq 0 \\ \Lambda_j(k) & \text{otherwise} \end{cases} \tag{13}$$

where

$$N_i(k) = \begin{cases} \|\bar{r}_i(k)\|^{-1} & \text{if } \bar{r}_i(k) \neq 0 \\ 1 & \text{otherwise} \end{cases}, \quad \text{and}$$
$$\bar{r}_j(k) := E[r_j(k)|Z^{k-1}]$$

In the hybrid state estimation block, the RMIMM algorithm is used to maintain the continuous-state estimate $\hat{x}$, its covariance P, and the mode estimate $\hat{m}$, for each of the N targets being tracked.

C. Identity Management Block

The trajectories estimated by the hybrid state estimation block do not clearly show the uncertainty about the identities of targets accumulated from interactions among crossing or nearby targets. In this section, we consider identity management, which entails the assignment of labels (or identities) to targets and the evolution of these labels over time. As long as the targets remain far apart, this problem is easily solved. This is easily seen in air traffic surveillance scenarios, in which air traffic controllers can typically distinguish between aircraft when they stay far apart, and have not interacted with each other. However, the interaction of multiple targets, especially those aircraft without transponders, which automatically transmit their own identities, makes the problem complex. A common approach to solving this problem is to maintain probabilities of all possible permutations of identity-target assignments, at each time step in the system. The complexity of such an algorithm grows exponentially in time, and is not practical to implement with the current computational capabilities.

We first define two types of matrices that we will encounter while analyzing identity management algorithms.

Definition 2: A stochastic matrix is one whose columns are probability vectors, that is, every column sums to 1. Matrix B is stochastic if $\sum_i B_{ij} = 1, \forall\ j$.

Definition 3: A doubly stochastic matrix is a stochastic matrix each of whose rows also sums to 1. Matrix B is doubly stochastic if $\sum_i B_{ij} = 1, \forall\ j$ and $\sum_j B_{ij} = 1, \forall\ i$.

A scalable algorithm for computing and maintaining multiple-target identity information had been proposed for use for multiple-target tracking in sensor networks [5,28]. This algorithm maintains identity information over time, when given information about the interaction between the N targets, in the form of the marginal probability distribution of target-identity associations. This information is stored in an $N \times N$ matrix $B(k)$, known as the *identity belief matrix*, where k is the current time step. The entry $B_{ij}(k)$ represents the probability that target j can be identified as label i. Because it represents the marginal probability distribution of target-identity associations, the belief matrix is doubly stochastic.

The evolution of this belief matrix is governed by a *mixing matrix* $M(k)$, whose element $M_{ij}(k)$ represents the probability that target i at time $k-1$ has become target j at time k. The belief matrix is updated as [5]

$$B(k) = B(k-1)M(k) \tag{14}$$

In certain applications, identity information about a target could be obtained from sensors that can measure its physical attributes, such as the shape and noise characteristics. For example, in applications of ad-hoc sensor networks, vision sensors (acoustic sensors) can measure the shape (noise characteristics) of a target and from this measurement, its identity can be inferred [29]. This information about the identity of a target obtained from sensors is called *local information* in this paper. Identity management can also use this target attribute information, if available from local sensors, to maintain the identity of a target correctly.

In this section, we consider a problem of how local information, if available, could be used to reduce the uncertainty of the belief matrix. For this, we use the statistical entropy (or Shannon information) of a probability vector $f \in [0,1]^n$, defined as

$$H(f) := \sum_{i=1}^{n} -f_i \ln f_i$$

as an uncertainty measure of the probability vector f. Using this, we define the average entropy of the belief matrix $B(k)$ of the N targets as

$$\bar{H}[B(k)] := \frac{1}{N}\sum_{j=1}^{N} H[b_j(k)] \tag{15}$$

where $B(k) = [\, b_1(k) \quad \cdots \quad b_N(k)\,]$. The average entropy of the belief matrix is used as a measure of the uncertainty in the identity of the N targets. In the belief matrix, because the columns represent the probabilities of identity belief for each target, the probability distribution of belief for each target is given by the corresponding column. Using this definition, we can prove the following Lemma 1:

Lemma 1: Let $\bar{H}[B(k)]$ be the average entropy over all the columns of the belief matrix $B(k)$. Then, $\bar{H}[B(k)] \geq \bar{H}[B(k-1)]$, if $B(k) = M(k)B(k-1)$; that is, mixing does not decrease the average entropy.

Proof: From the definition of average entropy of the system,

$$\begin{aligned}
\bar{H}[B(k)] &= \frac{1}{N}\sum_{j=1}^{N} H[b_j(k)] = \frac{1}{N}\sum_{j=1}^{N} H\{[M(k)B(k-1)]_j\} \\
&= \frac{1}{N}\sum_{j=1}^{N} H\left(\left[\sum_{i=1}^{N!}\alpha_i \Psi_i B(k-1)\right]_j\right) \\
&= \frac{1}{N}\sum_{j=1}^{N} H\left(\sum_{i=1}^{N!}\alpha_i [\Psi_i B(k-1)]_j\right) \\
&\geq \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N!}\alpha_i H\{[\Psi_i B(k-1)]_j\}
\end{aligned} \tag{16}$$

where Ψ_i is a permutation matrix. But premultiplying by a permutation matrix simply permutes the rows, so the set of values in the column does not change:

$$H\{[\Psi_i B(k-1)]_j\} = H[b_j(k-1)] \tag{17}$$

Therefore, we get

$$\begin{aligned}
\bar{H}[B(k)] &\geq \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N!}\alpha_i H[b_j(k-1)] = \frac{1}{N}\sum_{j=1}^{N} H[b_j(k-1)], \\
&\text{because } \sum_{i=1}^{N!}\alpha_i = 1 = \bar{H}[B(k-1)]
\end{aligned} \tag{18}$$

□

Corollary 1: Because $\bar{H}[B(k)] = \frac{1}{N}\sum_{j=1}^{N} H[b_j(k)]$ (sum over columns) $= \frac{1}{N}\sum_{j=1}^{N} H[b_j(k)]$ (sum over rows), the same proof of no decrease of entropy holds for mixing of the form $B(k) = B(k-1)M(k)$.

In the identity management algorithm, we assume that local information arrives in the form of column updates to the corresponding columns of the belief matrix. The column of the belief matrix corresponding to that particular target is replaced by the local information, which is the marginal probability distribution of the identity of the target. We preserve that specific column and scale the rest of the belief matrix to make it doubly stochastic. We compute this doubly stochastic matrix using a process called *Sinkhorn scaling*, which iteratively scales columns and rows of a given matrix; a detailed description, an analysis, and the rationale behind choosing this technique can be found in [30]. We now investigate in detail when the local information should be used to reduce uncertainties of the identity of the targets.

There are two possible forms of local information, depending on the level of certainty in the information. The first kind occurs when the sensor identifies the target with certainty, and the second occurs when the observation is in the form of a distribution. This can happen, for example, when signal processing is used to identify a target, and a statistical footprint of the possible identities of the target is obtained.

1) *Identity-type local information*: This is local information that gives with certainty the identity of one of the targets. In the implementation, this corresponds to local information in the form of a column unit vector.

2) *General forms of local information*: In general, local information is in the form of a stochastic column vector, that is, a vector whose elements sum to 1. In this case, clearly, the effect on the statistical entropy depends on the probability distribution that is observed as local information, and need not necessarily decrease the entropy. Consider, for example, the belief matrix

$$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

for two targets. The average entropy of this matrix is 0.5004. If local information arrives at column 2 in the form

$$\begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$$

(corresponding to information that target 2 has identity 2 with 70% probability), then the corresponding doubly stochastic matrix after Sinkhorn scaling is

$$\begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$$

The average statistical entropy of the updated matrix is 0.6109, that is, the statistical entropy increases when we incorporate information of this form.

The preceding statements have important implications in the incorporation of local information. We know that if the system were conducive to Bayesian normalization, then the average entropy of the system could only decrease with the incorporation of local information [5]. Because it is computationally quite simple to compute the average entropy, we only incorporate general local information if the doubly stochastic matrix after the Sinkhorn scaling has a smaller average entropy than before the incorporation of the local information.

The algorithm described in this section is the core of the MTIM algorithms for scenarios with clutter and without clutter, which will be presented in detail in the following two sections.

III. Environments with Clutter

In this section we consider a case in which there are extraneous measurements or clutter. Figure 4 shows the structure of the algorithm, and the evolution of hybrid state estimates and identity beliefs through a single time step. In particular, we draw attention to the difference between this figure and the algorithmic framework that was shown in Fig. 1, namely, that the hybrid state estimation component has been divided into two modules, the state prediction block and the state update block as shown in Fig. 4.

The following sections detail each block and present the logic behind various algorithmic choices.

A. State Prediction

State prediction is carried out for each of the N targets independently. For each target, the inputs are the state estimates from the previous time step, and the outputs are the state predictions and their covariances. The details that follow refer to the procedure used for a single target.

This block takes as input the continuous-state estimates $\hat{x}_i(k-1|k-1)$, covariances $P_i(k-1|k-1)$, and mode probabilities $\mu_i(k-1)$ from the previous time step $k-1$, where i refers to the mode of the target. The output of the block is a prediction of the continuous state and its covariance at time k without information from time k.

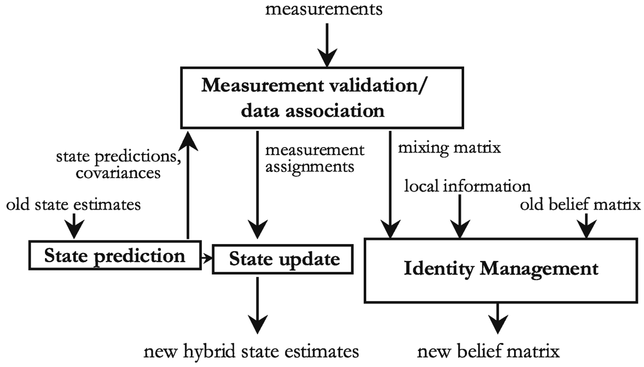


Fig. 4 Block diagram, showing a single time step of the multiple-target tracking and identity management algorithm for cluttered environments.

First, the mixing stage of RMIMM is used to combine the state estimates from the different modes, resulting in new combined initial states $\hat{x}_{0i}(k-1|k-1)$ and covariances $P_{0i}(k-1|k-1)$. These are inputs to a set of Kalman filters, tuned to each mode. This block corresponds to just the state prediction step of each Kalman filter. The outputs of the Kalman filters at this stage are the state predictions, their covariances, and the residual covariances

$$\begin{aligned} \hat{x}_i(k|k-1) &= A_i\hat{x}_{0i}(k-1|k-1) \\ P_i(k|k-1) &= A_iP_{0i}(k-1|k-1)A_i^T + Q_i \\ S_i(k) &= C_iP_i(k|k-1)C_i^T + R_i \end{aligned} \tag{19}$$

where i is the mode corresponding to each Kalman filter.

The mode estimate $\hat{m}(k-1)$ from the previous time step is used to obtain a single continuous-state prediction $\hat{x}(k|k-1)$ and a single residual covariance $S(k)$, corresponding to the aircraft staying in that mode. Because the predicted state is assumed to have a Gaussian distribution, the state prediction is the mean (center) of the ellipsoidal validation gate of the target, whereas the residual covariance is the covariance of the validation gate as shown in Fig. 2. Therefore, $S(k)$ would be expected to determine the size of the validation gate, according to (1). However, if the aircraft changes its mode (starts a maneuver), the Kalman filter has overestimated its confidence in its state estimate by using the mode estimate from the previous time step, which results in a smaller $S(k)$ than is appropriate. As a result, the measurement of the maneuvering target frequently does not fall inside its validation gate; therefore, the size of the validation gate must be increased. This increase is obtained by increasing the state covariance $S(k)$ with an additional term that accounts for the additional uncertainty that arises from a maneuvering target. This additional term depends on the state velocity estimate, $\hat{v}(k)$, and is given by the expression

$$S_{\text{extra}}(k) = \tau^2\hat{v}(k)\hat{v}(k)^{\mathrm{T}} + \nu^2\hat{v}_\perp(k)\hat{v}_\perp(k)^{\mathrm{T}} \tag{20}$$

where $\hat{v}_\perp(k)$ is obtained by rotating $\hat{v}(k)$ by 90 deg in the counterclockwise direction. The effective residual covariance $S'(k)$ is then equal to

$$S'(k) = S(k) + S_{\text{extra}}(k) \tag{21}$$

Because $S_{\text{extra}}(k)$ is positive definite, the region covered by the validation gate created from $S'(k)$ is larger than that created by $S(k)$, as shown in Fig. 5. In this figure, the smaller ellipse is the validation gate as determined by $S(k)$, whereas the larger ellipse is that

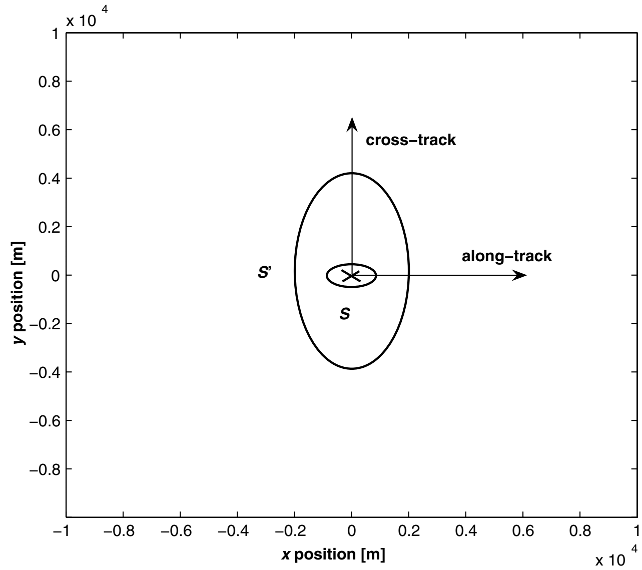


Fig. 5 Validation gates determined by the original residual covariance S and the effective residual covariance S', which accounts for the maneuvering uncertainty of a target.

determined by $S'(k)$. The constants τ and ν are chosen empirically to ensure that maneuvers are unlikely to lead to measurements outside validation gates, and the cross-track term ν is chosen to be larger than the along-track term τ. The term S_{extra} is related to the velocity estimate of the target because errors in track due to a change in the flight mode are dependent on the velocity of the target. The outputs from the state prediction block are therefore the predicted state $\hat{x}_t(k|k-1)$, its residual covariance $S_t(k)$, and effective residual covariance $S_t'(k)$ for target t. There are N such sets of outputs, one set for each target being tracked. The effective residual covariance $S'(k)$ is used only for measurement validation purposes.

B. Measurement Validation/Data Association

We first define the residual $r_j^t(k) = z_j(k) - C^t\hat{x}^t(k|k-1)$ for target t and measurement j. The measurement at position $z_j(k)$ falls inside the validation gate if (1) is satisfied, with the residual covariance $S(k)$ replaced by the effective residual covariance $S'(k)$. As in Sec. III.A, measurements that fall within the validation gate of one of the targets are called validated measurements. Therefore, the first step is to use the validation gate to choose the validated measurements. Given the set of validated measurements, a modified, approximate version of JPDA is used to obtain the association probability matrix $\beta(k)$ and the mixing matrix $M(k)$.

The approximate JPDA algorithm is computationally more tractable than JPDA [10]. As we saw in Sec. II.A, the JPDA algorithm requires computing the sum of a joint probability distribution over $\frac{m_k!}{(m_k-N-1)!}$ events, which can be very large in cluttered environments, where m_k is potentially large. Formulating all possible joint events and summing over them is therefore quite tedious [21], given the computational limitations of most sensor networks. Even in the context of air traffic surveillance, this task is difficult to perform in real time in highly cluttered environments. This is the motivation for approximate data association methods such as the approximate JPDA.

Let us denote the Gaussian probability density function of the residual

$$G_{jt}(k) := \mathcal{N}_t[z_j(k)]$$

Therefore, $G_{jt}(k)$ is proportional to the Gaussian likelihood function that represents the closeness between target t and measurement j. We let

$$P_{\text{st}}(k) := \sum_{j=1}^{m_k} G_{jt}(k), \quad \text{and} \quad P_{rj}(k) := \sum_{t=1}^{N} G_{jt}(k) \tag{22}$$

Then, the marginal association probability that was described in (6) can be approximated by the following expression [10]:

$$\beta_{jt}(k) = \frac{G_{jt}(k)}{P_{\text{st}}(k) + P_{rj}(k) - G_{jt}(k) + B_{\text{bias}}} \tag{23}$$

In short, (23) places a greater weight on a target that does not fall into the validation gates of any of the other targets than the weight placed by the JPDA algorithm on such a target. In the preceding expression, B_{bias} is a bias term that is set to zero in most cases, including in all the examples presented in this paper.

Unlike the association matrix produced by the JPDA algorithm, the data association matrix computed by the approximate JPDA algorithm is not necessarily a stochastic matrix and thus it does not satisfy a constraint that there is only one measurement from one target. As a result, the accuracy of approximate JPDA may not be suitable for certain situations. To remedy this, and to improve the performance of data association, we propose a modification of the approximate JPDA algorithm, which uses the Sinkhorn algorithm [30,31] to make the data association matrix $\beta(k)$ doubly stochastic, so that it can be used as the mixing matrix for identity management. We have shown in prior work [32] that the Sinkhorn algorithm would minimize a probabilistic distance from the prior constraint-violating matrix. The modified algorithm keeps the essential characteristics of the JPDA algorithm with far less computational complexity than JPDA, while tracking many targets in clutter. We refer to this algorithm (the combination of the approximate JPDA and Sinkhorn algorithms) as the modified approximate JPDA (MAJPDA) algorithm.

Because there are more measurements than targets in a cluttered environment, it is necessary to choose a square submatrix of the data association matrix as the mixing matrix. The MAJPDA algorithm that we propose involves determining both the association probability matrix and a doubly stochastic, square mixing matrix.

For an environment with clutter, if there are L measurements, the association matrix β has L rows and $N \le L$ columns. The mixing matrix $M(k)$ must still be a $N \times N$ matrix. To choose N rows from among the L possible rows, we use the extended Munkres algorithm [16].

The extended Munkres algorithm lends itself to processing the marginal association probability matrix output by the proposed MAJPDA. The N numbers chosen from $\beta(k)$ by the extended Munkres algorithm correspond to the measurements $z_j(k)$, which are the N measurements assigned to the N targets so as to maximize the sum of association probabilities. The assignment of measurements to targets is a one-to-one correspondence between measurement j and target t, that is, j is a function of t, and vice versa. The N rows of $\beta(k)$ representing these measurements form an $N \times N$ matrix. The doubly stochastic form of this matrix serves as the mixing matrix $M(k)$. Then, the data association algorithm can be described as follows:

Algorithm 1: Data Association Algorithm

Given: validated measurements $z_j(k)$ ($j = \{1, \cdots, L\}$) and targets t ($t \in \{1, \cdots, N\}$) where $L \ge N$.

1) Modified approximate JPDA (MAJPDA)
   a) Compute the $L \times N$ association probability matrix $\beta'(k) = [\beta'_{jt}(k)]$ using (23).
   b) Scaling: Find $\beta(k) = \text{SI}[\beta'(k)]$ such that $\sum_{j=1}^{L} \beta_{jt} = 1$ and $\sum_{t=1}^{N} \beta_{jt} = 1$ where the operator SI represents the Sinkhorn scaling process.
2) Measurement assignments (extended Munkres algorithm): Find a permutation Π such that

$$\max_{\Pi(t)} \quad \sum_{t=1}^{N} \beta_{\Pi(t)t} \quad \text{subject to} \quad 1 \le t \le N$$
$$1 \le \Pi(t) \le L \qquad i \ne j \Rightarrow \Pi(i) \ne \Pi(j)$$

3) Mixing matrix: $M(k) = \text{SI}(\beta_{\Pi(t)t})$ for $t \in \{1, \cdots, N\}$.

The mixing matrix and measurement assignments are then passed to the belief matrix update and state estimate update blocks, respectively.

C. State Estimate Update

Given a measurement for target t, namely, $z_j(k)$, the state update step of the RMIMM algorithm propagates the estimate of the continuous state $\hat{x}^t(k-1|k-1)$, its covariance $P^t(k-1|k-1)$, and the mode probabilities $\mu^t(k-1)$ to time k. The outputs of the state estimate update block at time k include $\hat{x}^t(k|k)$, $P^t(k|k)$ and $\hat{m}^t(k)$. These outputs are then used as inputs to the target tracking and identity management algorithm at the next time step, $k+1$.

D. Identity Management

1. Mixing Matrix

The mixing matrix $M(k)$ from the measurement validation and data association block is input to the belief matrix update block. The evolution of the belief matrix is governed by (14). The mixing matrix is equal to the identity matrix if none of the targets interacts with another, that is, there is no uncertainty in the identities of the targets. The belief matrix at time k is maintained independently of the hybrid state estimation component of the algorithm, and is only used to compute the updated belief matrix at the next time step. Apart from being changed during mixing events in which two targets move close to each other, the belief matrix can also be altered when local information is received.

2. *Local Information Incorporation*

Local information is useful only if its incorporation results in the uncertainty of the belief matrix being reduced, where uncertainty is measured as the statistical entropy of the belief matrix as discussed in Sec. II.C.

In this section, we propose an additional source of possible local information. The proposed information is automatically generated whenever targets interact and the entropy of the belief matrix increases significantly. Without using extra sensors to get attribute information about the targets to correct target identities, we propose the use of the multiple hypothesis testing (MHT) algorithm to infer the local information about interacting targets. The reason for using MHT is that it uses all the measurements over a time period and considers all possible target-identity association hypotheses over that period. Thus, it could provide more accurate target-measurement association, but its computational complexity grows exponentially as the number of time steps considered increases. In our implementation, the MHT algorithm is used only when the probability of the most likely identity association for a target is below a preset threshold, which we treat as a design parameter.

The local information about the target identities is derived by applying the MHT algorithm on the state estimates across two time steps (or more steps depending on the difficulty of resolving uncertainty) in this section. Such information is useful in situations such as the one shown in Fig. 6. In this figure, two aircraft cross at right angles to each other at time k. Their estimated positions are marked with "x"s, whereas the radar measurements are marked with "o"s. The expression $\hat{x}^t(k)$ denotes the state (position) estimate for aircraft t at time k. The measurements $z_a(k)$, $z_b(k)$, $z_c(k+1)$, and $z_d(k+1)$ are indexed by letters to reflect the fact that from many possible choices, two measurements are chosen by MAJPDA at each time step to correspond to the two targets. Aircraft A starts at the southwest corner at time $k-1$ and moves to the northeast corner at time $k+1$, whereas aircraft B starts at the northwest corner and moves to the southeast corner. We assume that aircraft A (and, respectively, aircraft B) is target 1 (respectively, target 2) with absolute certainty at time $k-1$. In other words, the belief matrix $B(k-1)$ is the identity matrix, that is,

$$B(k-1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

At time k, the two targets are close together and almost equally likely to be associated with each of two measurements. For example, aircraft A is target 1 with probability of 0.51 and is target 2 with probability of 0.49. Then the belief matrix at time k is

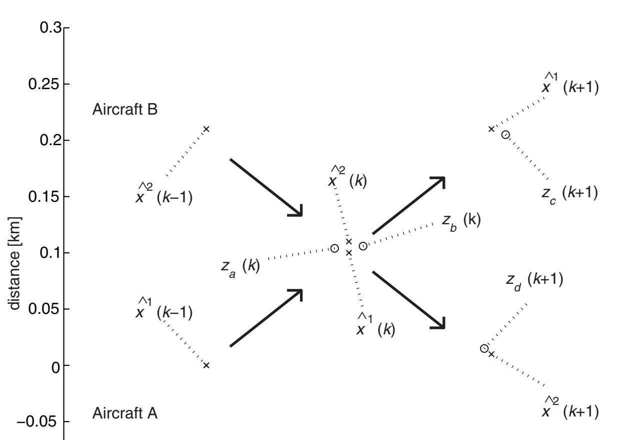


Fig. 6 State estimates (x) and measurements (o) for two-aircraft example. The solid arrows denote the direction of movement of the targets. The dotted lines do not denote distances; they give the association between the labels and the points.

$$B(k) = B(k-1)M(k) = \begin{bmatrix} 0.51 & 0.49 \\ 0.49 & 0.51 \end{bmatrix}$$

At time $k+1$, the aircraft tracks have diverged, and the validation gates of the two aircraft no longer intersect. Therefore, the mixing matrix $M(k+1)$ computed by the proposed MAJPDA (Sec. III) is the identity matrix, and the belief matrix at time $k+1$ stays the same, that is,

$$B(k+1) = B(k)M(k+1) = \begin{bmatrix} 0.51 & 0.49 \\ 0.49 & 0.51 \end{bmatrix}$$

The MAJPDA algorithm cannot differentiate between the two measurements at time k. As a result, the uncertainty (as measured by the statistical entropy) of the belief matrix is essentially maximum. This uncertainty remains even after the aircraft separate. This is an inherent problem in multiple-target tracking, when aircraft move close to each other. If there were no identity management taking place, but only JPDA for multiple-target tracking, the marginal association probabilities in this scenario would be 0.5 for all associations, and there would be a 50% chance of target swapping. The advantage of identity management is that we explicitly maintain beliefs of the identities, and can employ corrective measures through local information when necessary. In other words, if the aircraft model could have followed the correct target, then the belief matrix would not have become uncertain through mixing.

As we will see next, by simply analyzing the dynamics of the two aircraft, a belief matrix with lower entropy can be determined. The MHT algorithm is used to obtain a belief matrix with lower entropy than can be achieved by just MAJPDA and standard belief matrix updates. The MHT algorithm is discussed in detail in [2,18]. Given the initial conditions $\hat{x}^1(k-1)$ and $\hat{x}^2(k-1)$, as well as the measurements $z_a(k)$, $z_b(k)$, $z_c(k+1)$, and $z_d(k+1)$, there are four possible target-measurement matches that can occur; these are illustrated in Fig. 7. Figure 7a refers to the outcome chosen by MAJPDA, because target 1 is assumed to have gone through measurements $z_a(k)$ and $z_c(k+1)$. Each plot in Fig. 7 is a representation of a joint event, composed of the four events represented by the line segments in the plot. The likelihood of the joint event that each target actually corresponds to the pair of measurements associated with it, is equal to the product of the likelihood of the individual events. The result is four likelihoods for the four joint events portrayed in the plots of Fig. 7.

To determine the belief matrix, one is only interested in whether or not target 1 reaches the expected position of aircraft A at time $k+1$. Therefore, the sum of the likelihoods of the events shown in Figs. 7a and 7c is the likelihood that target 1 remains identified as aircraft A, as well as the likelihood that target 2 remains identified as aircraft B; we denote this quantity l_1. The sum of the likelihoods from Figs. 7b and 7d is the likelihood that the targets swap identities; let this quantity be denoted l_{-1}. Based on the dynamics of commercial jets that we consider in this example, the maneuvers in Figs. 7b and 7d are highly unlikely so that MHT computes $l_{-1} = 0$.

The doubly stochastic version of the matrix

$$\begin{bmatrix} l_1 & l_{-1} \\ l_{-1} & l_1 \end{bmatrix}$$

represents the mixing matrix for the two aircraft between time steps $k-1$ and $k+1$. This matrix, a two-step mixing matrix, is denoted as $\Gamma(k+1)$. For the example we have just seen, $\Gamma(k+1)$ is the identity matrix. Therefore, the belief matrix determined by MHT at time $k+1$ is

$$B'(k+1) = B(k-1)\Gamma(k+1) = \begin{bmatrix} l_1 & l_{-1} \\ l_{-1} & l_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The resulting belief matrix $B'(k+1)$ is the identity matrix, which has lower entropy than $B(k+1)$ that would have been obtained from the standard multiple-target tracking and identity management algorithm model described in Sec. II. The local information can

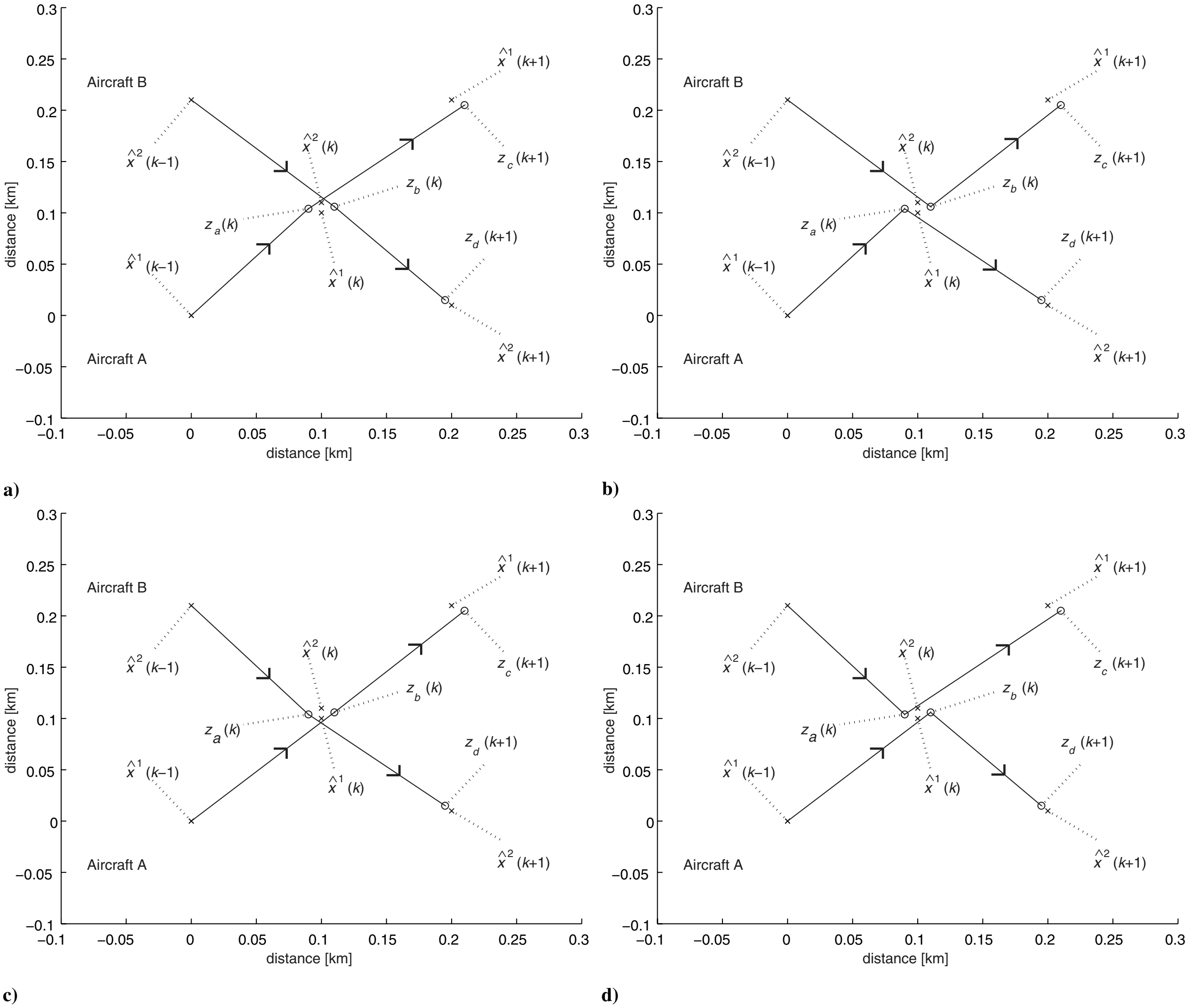


Fig. 7 Possible joint events in MHT. The solid arrows denote the direction of movement of the targets. The dotted lines do not denote distances; they give the association between the labels and the points.

therefore be incorporated through the *identity management* block (Fig. 4).

This example is an ideal case in which the new belief matrix has no uncertainty, i.e., MHT computes the identity matrix. In general the MHT module attempts to use knowledge of the dynamics of the aircraft to resolve uncertainty in target-identity associations probabilistically and thus the belief matrix is a doubly stochastic matrix. Knowing the Markovian probability of discrete mode transitions and looking two time steps back, the MTH module can compute the probability of target-identity associations to use as local information. Because there is no guarantee that the MHT local information computed earlier in this paper will always improve the entropy of the belief matrix, it is only incorporated if this local information decreases the entropy. This automatic computation of local information and its incorporation are both carried out before the information is sent to the identity management component of the multiple-target tracking and identity management algorithm (Fig. 4).

We note that the framework we introduce in this paper is flexible enough to accommodate many variations and extensions. For example, because target-measurement association in the MTIM algorithm is based on MAJPDA, it has the same limitations as JPDA; it cannot perform track initiation and termination and the number of targets is assumed to be known and constant. However, by replacing the measurement validation/data association block with other algorithms that can deal with the aforementioned limitations, we can extend the performance of the MTIM algorithm. We have recently developed a decentralized version of the MTIM algorithm that can address some of the preceding limitations [8,19,20].

We now present some simulations to demonstrate how the proposed techniques for target tracking and identity management could be implemented, taking into account the practical limitations of the radar-based sensors that are being used for air traffic surveillance.

E. Simulations

Two examples are presented in this section to demonstrate the efficacy of the MTIM algorithm in clutter. Both examples are scenarios in which multiple aircraft are interacting in a cluttered environment. In both examples, several system parameters are set to the same values. First, measurement points are made available every 5 s. Measurement covariance R is

$$\begin{bmatrix} (100)^2 & 0 \\ 0 & (100)^2 \end{bmatrix}$$

which means the standard deviation of position error is 100 m in both dimensions. Process noise covariance Q is set to be

$$\begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$$

for straight flight and

$$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

for the turning mode. The large process noise covariance for the turning mode represents the uncertainty in the transient motion of an aircraft, yet the straight flight has a small covariance representing the steady-state motion of the aircraft. The preceding constants are realistic values for aircraft in clutter and are taken from [34]. Clutter is uniformly distributed in space and Poisson-distributed in number; the density of clutter points is 0.5×10^{-6} clutter points per square meter, or 0.5 points per square kilometer. The validation gate parameter γ is set to 9.2, which would correspond to a 3σ confidence level if residual covariance S were used. The effective residual covariance S' that is actually used is determined with system constants τ and ν set to 3 and 6, respectively. The threshold for initiating MHT is set to 0.75. In both examples, the target 1 is initially identified as aircraft A, 2 as B, and so on.

The first example is designed to demonstrate the effectiveness of the MTIM algorithm in a realistic free-flight scenario. There are six aircraft flying in both straight and turning modes. We show the measurements obtained with clutter in Fig. 8 (top). The trajectory plot in Fig. 8 (center) is set up the same way as that in Fig. 9 (center). This plot includes realistic accident scenarios. For example, the intersection of aircraft A and B at coordinates (20, 10) depict a blunder by aircraft B into aircraft A's path. Tracking of six aircraft is successful except for overshoot at the start of a maneuver. The dashed lines for the target measurements are visible at these overshoot points. Identity management also performs well; part of the belief matrix is shown in Fig. 8 (bottom). Only the beliefs for the first two targets are shown. One can see the interaction between aircraft A and B at $k = 21$, which leads to an increase in the statistical entropy in the belief matrix followed by restoration of the belief matrix through automated local information incorporation using MHT. Aircraft A and F have a similar interaction at $k = 37$; only the changes in the belief matrix for target A are shown in Fig. 8 (bottom). The interaction between aircraft B and C at $k = 27$ is of note because the interaction is mild. The belief matrix is not changed enough to trigger the automated local information, so the belief of target 2 is not restored. Because the interaction is mild, one can confidently label target 2 as aircraft B. The identity management portion of MTIM performs successfully.

The next example is an extreme (aerobatic) scenario in which four aircraft fly at each other directly and maneuver; this example is useful in understanding the capabilities of MTIM. Figure 9 (top) shows a shot of the radar screen including the entire flight data, but without the trajectories explicitly indicated. This gives us an idea of the clutter density, as well as how unclear the system is, especially when the aircraft come close to each other. Figure 9 (center) displays the actual and estimated positions of four aircraft following symmetric paths that first converge, then maneuver around a common point, and finally diverge. The dashed lines with dots as markers are the noisy measurements from the targets. The solid lines with markers as shown in the legend are the estimated positions found by MTIM. The fainter dots interspersed throughout the plot are clutter points. Aircraft A, B, C, and D fly with constant velocity of 200 m/s. All turns are executed at 3 deg/s. Target tracking is accurate except for overshoot when aircraft start turning. Indeed, the dashed lines depicting the noisy measurements are not clearly visible because the solid lines depicting estimated target positions match them almost exactly. Figure 9 (bottom) displays the evolution of the belief matrix in graphical form. The plots, from top to bottom, show the probability that any aircraft is identified with targets one through four, respectively. From this figure, it is clear that the belief matrix is unchanged while the aircraft are distant from and not interacting with each other. When paths cross, the belief matrix is changed significantly only if the measurements for both targets happen to nearly coincide. For example, at $k = 30$, targets 1 and 2 nearly coincide, leading to the belief that both targets 1 and 2 are nearly 0.5 aircraft A and 0.5 aircraft B. However, the automated MHT local information generated by this interaction restores the belief matrix to nearly identity at the following time step. At $k = 30$, targets 3 and 4 also interact with equally drastic loss of identity between aircraft C and D. Again, the local information restores the belief matrix at the following time step. At $k = 32$, targets 1 and 3 interact, with similar jump in belief matrix entropy followed by belief matrix restoration from local information. Targets 2 and 4 also interact in the same fashion at $k = 32$. The scenario depicted in Fig. 9 establishes the

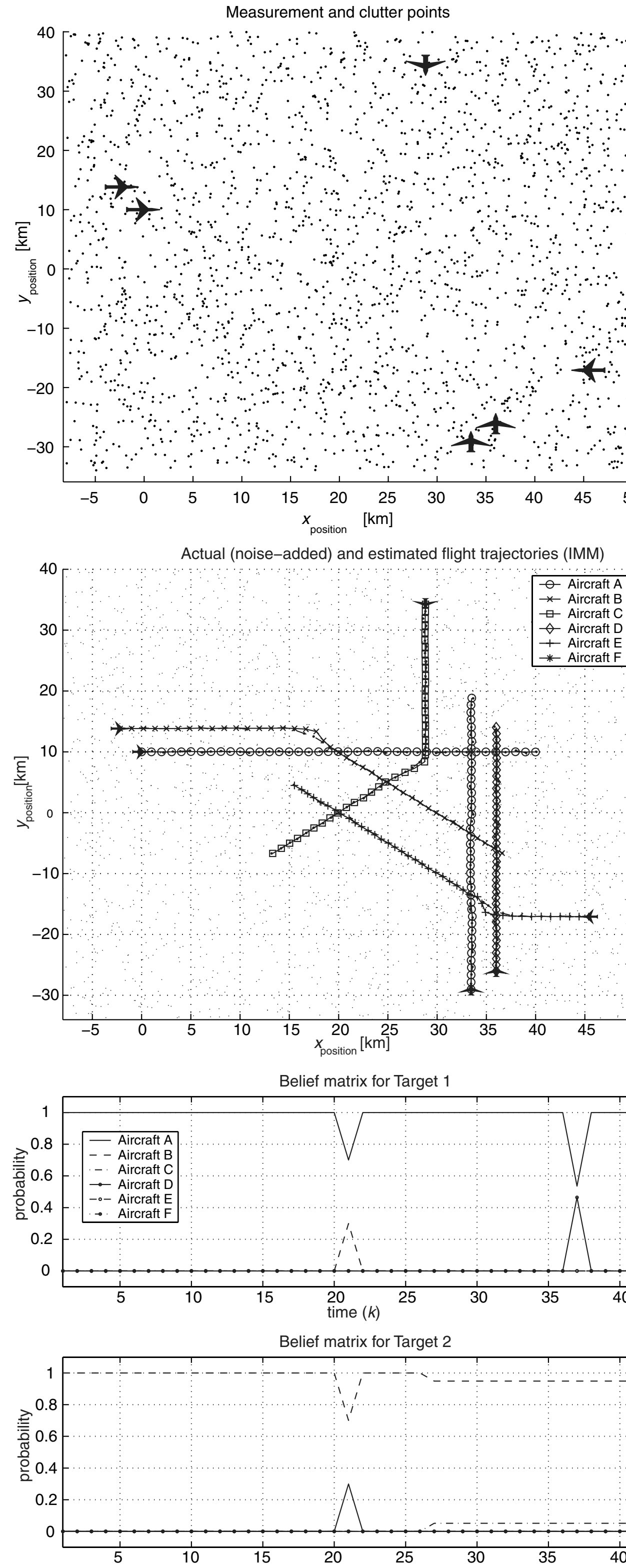


Fig. 8 Measurement points with clutter (top), aircraft trajectories (center), and plot of belief information for aircraft A and B (bottom).

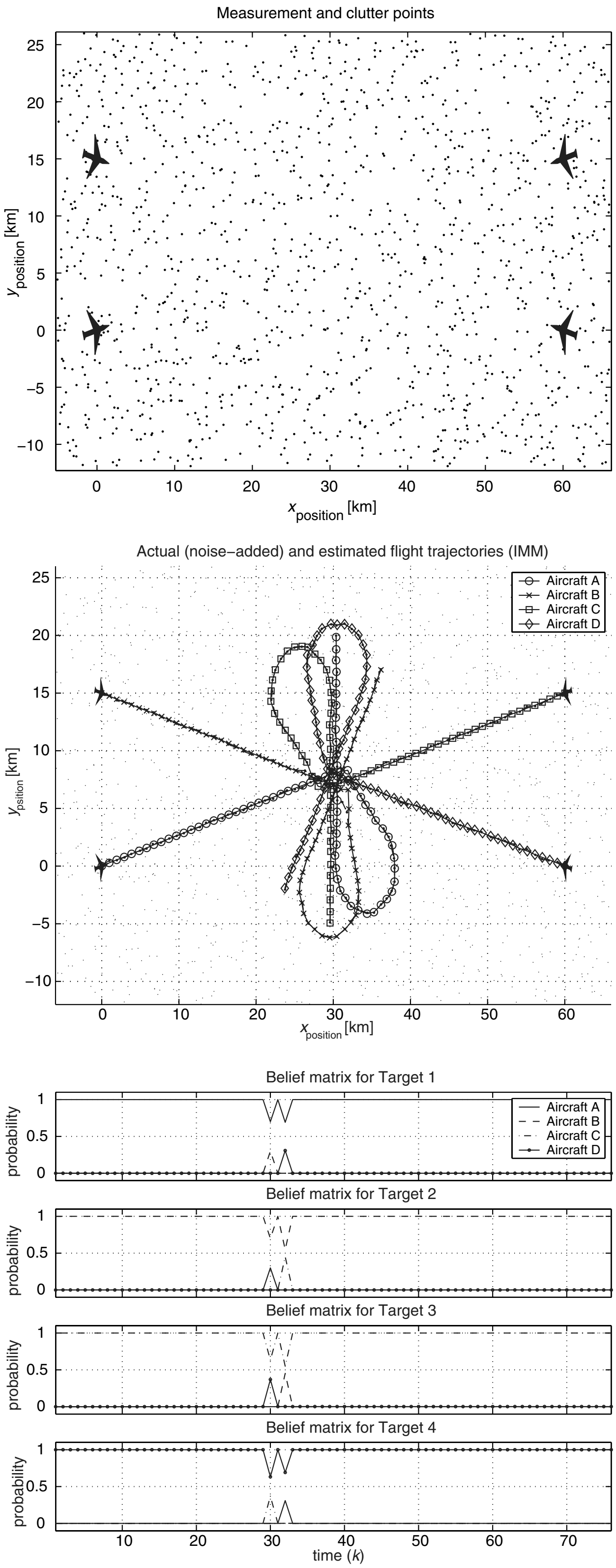


Fig. 9 Measurement points with clutter (top), aircraft trajectories (center), and accompanying belief matrix plot (bottom).

efficacy of the MTIM algorithm in clutter. Indeed, MTIM performs successfully for both examples shown.

IV. Environments Without Clutter

In this section, we consider a special case in which there are no clutter or undetected targets. In the absence of clutter, if all targets are detected, there are N targets and N measurements at every time step; in other words, the target detection probability $P_D = 1$, the number of unassociated measurements $\phi(\Theta) = 0$, and the measurement association indicator $\tau_j(\Theta) = 1$. Then, the marginal association probability, given by (6), becomes

$$\beta_{jt}(k) = \sum_{\Theta}\left(\prod_{i=1}^{N} \mathcal{N}[z_i(k)]\hat{\omega}_{jt}(\Theta)\right) \tag{24}$$

The tracking algorithm is the same as that described in Sec. II.B. The mixing matrix, described in (14), represents the degree of mixing of identities between interacting targets. Because the state estimate for target j at time k is computed in a deterministic manner using a hybrid estimation algorithm, given the measurement associated with target j at time k, the probability $M_{ij}(k)$ is the same as the probability that the measurement associated with target j at time k (let us denote this measurement t) corresponds to the state estimate of target i at time $k-1$. Indeed, this is equal to the association probability $\beta_{it}(k)$. Therefore, ideally, we would use the association matrix given by $\beta(k)$ in (24), as the mixing matrix. However, as we saw in Sec. II.C, the mixing matrix needs to satisfy the physical constraint of being doubly stochastic, a feature that is lost while computing the marginal association matrix. Using the Sinkhorn scaling process, we construct a doubly stochastic matrix $\beta'(k)$ that approximates $\beta(k)$, and we use $\beta'(k)$ as the mixing matrix $M(k)$ for the belief matrix updates. Then, the evolution of the belief matrix is governed by (14). Thus, the MTIM algorithm for an environment without clutter is as follows:

Algorithm 2: Multiple-target tracking and identity management without clutter

For target t ($t \in \{1, \cdots, N\}$) at time k,

Step 1: Mixing/interaction: initialize $\hat{x}_{0i}(k-1|k-1)$ and $P_{0i}(k-1|k-1)$ (for mode i, $i \in \{1, \cdots, N_{mode}\}$)

Step 2: Kalman filter i

1) State propagation/prediction

$$\begin{aligned}\hat{x}_i(k|k-1) &= A_i\hat{x}_{0i}(k-1|k-1)\\ P_i(k|k-1) &= A_iP_{0i}(k-1|k-1)A_i^T + Q_i\\ S_i(k) &= C_iP_i(k|k-1)C_i^T + R_i\end{aligned} \tag{25}$$

2) Measurement validation

$$r_{ij}(k)^T S_i(k)^{-1} r_{ij}(k) \leq \gamma^2 \tag{26}$$

where $r_{ij}(k) = z_j(k) - C_i\hat{x}_i(k|k-1)$ ($j \in \{1, \cdots, m_k^i\}$) and m_k^i is the number of validated measurements for target i at time k.

3) Measurement update

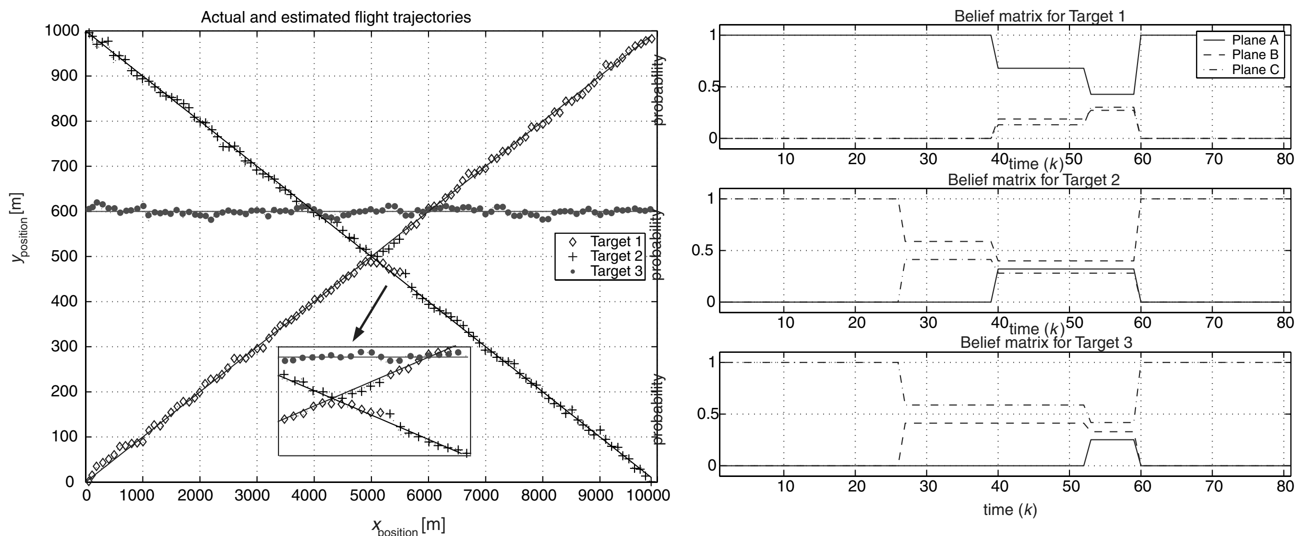
a) Compute an association matrix using (24) and a mixing matrix by making the association matrix a doubly stochastic matrix using the Sinkhorn scaling process.

b) Update the belief matrix using (14).

c) If local information arrives and it decreases the entropy of the belief matrix, then update the column corresponding to the local information, and scale the rest of the matrix using the Sinkhorn scaling process to make it doubly stochastic.

d) Update the continuous-state estimate and its covariance

$$\begin{aligned}\hat{x}_i(k|k) &= \hat{x}_i(k|k-1) + K_i(k)\sum_{l=1}^{m_k^i}\beta_{il}(k)r_{il}(k)\\ P_i(k|k) &= [I - K_i(k)C_i]P_i(k-1|k-1)\\ &\quad + K_i(k)\Bigg[\sum_{l=1}^{m_k^i}\beta_{il}(k)r_{il}(k)r_{il}(k)^T\\ &\quad - \left(\sum_{l=1}^{m_k^i}\beta_{il}(k)r_{il}(k)\right)\times\left(\sum_{l=1}^{m_k^i}\beta_{il}(k)r_{il}(k)\right)^T\Bigg]K_i(k)^T\end{aligned} \tag{27}$$



a) b)

Fig. 10 Tracking and identity management of three aircraft (no clutter): a) estimated aircraft trajectories, b) belief matrix evolution.

where a Kalman filter gain $K_i(k) = P_i(k|k-1) \times C_i^T [C_i P(k|k-1) C_i^T + R_i]^{-1}$.

Step 3: Compute mode likelihood functions

$$\Lambda_i(k) = \mathcal{N}[r_i(k); 0, S_i(k)] \tag{28}$$

where $r_i(k) := \sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k)$.

Step 4: Compute mode probabilities: $\mu_i(k)$.

Step 5: Compute outputs: $\hat{x}(k|k)$, $P(k|k)$, $\hat{m}(k)$ using (12) and $B(k)$. □

In this section, we consider a scenario in which the number of measurements at each time step equals the number of aircraft being tracked. We consider a three-aircraft scenario, in which, given the noisy measurements of the position coordinates of three different aircraft in random order at every instant of time, we track the aircraft not only in their state estimates, but also with respect to their identities. The aircraft fly straight with constant velocity. This is representative of a scenario in which we might receive local information about one of the aircraft, based on, perhaps, its physical attributes such as the shape and noise characteristics of the aircraft.

The simulation results are shown in Fig. 10. Initially, target 1 (2 and 3) is plane A (B and C, respectively). Figure 10b shows the evolution of the belief matrix. We notice that a decrease in belief corresponds to interactions between targets, due to their proximity. This is seen by comparing the trajectories and belief matrix plots in Figs. 10a and 10b respectively. The first interaction is between targets 2 and 3, and the belief of target 1 is unaffected. Target 2 then proceeds to interact with target 1, and the final interaction involves targets 1 and 3. We can see that there is target-swapping between targets 1 and 2 in Fig. 10a over a period of time. During this period, the maximum probability attains a value below 0.5, which implies that we do not have much confidence in those associations being the actual ones. In this simulation, local information of the identity type for target 1 is obtained at time 60, and results in an immediate improvement of the beliefs of all three targets. Using only local information (for one aircraft), the proposed algorithm can reduce uncertainties in the identities of all aircraft. It is evident from the figures that the algorithm we have proposed maintains an accurate estimate of the trajectories as well as the identities in the presence of multiple possible associations of measurements to targets, provided there are sources of local information, albeit intermittent ones.

V. Conclusions

We have developed an algorithmic framework for multiple-target tracking and identity management that can track and manage identities of multiple maneuvering targets simultaneously. This algorithm is composed of three different blocks: data association, tracking/hybrid state estimation, and identity management. For data association, we have proposed a modified approximate joint probability density algorithm that can handle many targets with low computational complexity, yet possesses the main advantages of the standard JPDA algorithm. For tracking multiple maneuvering targets, we used the residual-mean interacting multiple model algorithm based on the hybrid (or multiple) aircraft dynamics models. For identity management, we developed an algorithm that can keep track of the identities of targets over time probabilistically. The proposed algorithm could be used as a supporting tool for tracking and identifying aircraft in air traffic control. The MTIM algorithm could increase safety and decrease controllers' workload by providing accurate aircraft tracking and identity information even when the onboard transponders and/or secondary surveillance radars are malfunctioning and thus identity information is not available, or there are general aviation aircraft in the air that are not equipped with transponders. In military applications, the MTIM algorithm could be used for a passive identification friend or foe (IFF) system that does not need active interrogation so that aircraft do not expose their presence to the enemy.

Because the MTIM algorithm uses an approximate version of JPDA for data association, it has the same limitations as JPDA; it cannot perform track initiation and termination of an unknown number of targets. Based on the algorithmic framework proposed in this paper, we recently developed a decentralized algorithm that can address the aforementioned limitations. Our future research includes asynchronous updates between measurements and the performance improvement of multiple-target tracking by using identity information.

References

[1] Bar-Shalom, Y., and Fortmann, T., *Tracking and Data Association*, Academic Press, New York, 1988.

[2] Talotta, N. J., "A Field Study for Transponder Performance in General Aviation Aircraft," U.S. Department of Transportation Federal Aviation Administration, Tech. Rep. DOT/FAA/CT-97/7, December 1997.

[3] Nolan, M. S., *Fundamentals of Air Traffic Control*, 4th ed., Brooks Cole, Belmont, CA, 2003.

[4] Krause, S. S., *Avoiding Mid-Air Collisions*, TAB Books, New York, 1995.

[5] Shin, J., Guibas, L., and Zhao, F., "A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks," *Information Processing in Sensor Networks*, edited by F. Zhao and L. Guibas, Lecture Notes in Computer Science 2654, Springer, New York, April 2003, pp. 223–238.

[6] Chu, M., Mitter, S. K., and Zhao, F., "Distributed Multiple Target Tracking and Data Association in Ad Hoc Sensor Networks," *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Australia, July 2003, pp. 447–454.

[7] Clark, M., Maskell, S., Vinter, R., and Yaqoob, M., "A Comparison of the Particle and Shifted Rayleigh Filters in Their Application to a Multi-Sensor Bearings-only Problem," *Proceedings of IEEE Conference on Aerospace*, March 2005, pp. 1–6.

[8] Oh, S., Hwang, I., Roy, K., and Sastry, S., "A Fully Automated Distributed Multiple-Target Tracking and Identity Management Algorithm," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, San Francisco, Aug. 2005.

[9] Oh, S., and Sastry, S., "An Efficient Algorithm for Tracking Multiple Maneuvering Targets," *Proceedings of the IEEE Conference on Decision and Control*, Seville, Spain, Dec. 2005.

[10] Fitzgerald, R. J., "Development of Practical PDA Logic for Multitarget Tracking by Microprocessor," *Multitarget-Multisensor Tracking: Advanced Applications*, Vol. 1, edited by Y. Bar-Shalom, Artech House, Norwood, MA, 1990, pp. 1–23.

[11] Roecker, J., "A Class of Near Optimal JPDA Algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 30, No. 2, April 1994, pp. 504–510.

[12] Zhou, B., and Bose, N., "Multitarget Tracking in Clutter: Fast Algorithm for Data Association," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 2, April 1993, pp. 352–362.

[13] Hadzagic, M., Michalska, H., and Jouan, A., "IMM-JVC and IMM-JPDA for Closely Maneuvering Targets," *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, Pacific Grove, CA, Nov. 2001, pp. 1278–1282.

[14] Popp, R., Pattipati, K., and Bar-Shalom, Y., "Dynamically Adaptable M-Best 2-D Assignment Algorithm and Multilevel Parallelization," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 4, Oct. 1999, pp. 1145–1160.

[15] Munkres, J., "Algorithms for the Assignment and Transportation Problems," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 5, No. 1, March 1957, pp. 32–38.

[16] Bourgeois, F., and Lassalle, J., "An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices," *Communications of the Association for Computing Machinery*, Vol. 14, No. 12, Dec. 1971, pp. 802–806.

[17] Blom, H., and Bar-Sharlom, Y., "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Transactions on Automatic Control*, Vol. 33, No. 8, Aug. 1988, pp. 780–783.

[18] Blackman, S., *Multiple-Target Tracking with Radar Applications*, Artech House, Norwood, MA, 1986.

[19] Hwang, I., Roy, K. H. B., and Tomlin, C., "A Distributed Multiple-Target Identity Management Algorithm in Sensor Networks," *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 2004.

[20] Oh, S., Hwang, I., and Sastry, S., "Distributed Multiple-Target Tracking and Identity Management Algorithm," *Journal of Guidance, Control, and Dynamics* (to be published).

[21] Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M., "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," *IEEE Journal of Oceanic Engineering*, Vol. OE-8, No. 3, July 1983, pp. 173–184.

[22] Chang, K.-C., and Bar-Shalom, Y., "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers," *IEEE Transactions on Automatic Control*, Vol. 29, No. 7, July 1984, pp. 585–594.

[23] Chang, K.-C., Chong, C.-Y., and Bar-Shalom, Y., "Joint Probabilistic Data Association in Distributed Sensor Networks," *IEEE Transactions on Automatic Control*, Vol. 31, No. 10, Oct. 1986, pp. 889–897.

[24] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J., "Interacting Multiple Model Methods in Target Tracking: A Survey," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 1, Jan. 1998, pp. 103–122.

[25] Hwang, I., Balakrishnan, H., and Tomlin, C., "State Estimation for Hybrid Systems: Applications to Aircraft Tracking," *IEE Proceedings: Control Theory and Applications*, Vol. 153, No. 5, Sept. 2006, pp. 556–566.

[26] Hwang, I., Hwang, J., and Tomlin, C., "Flight-Mode-Based Aircraft Conflict Detection Using a Residual-Mean Interacting Multiple Model Algorithm," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug. 2003.

[27] Balakrishnan, H., Hwang, I., Jang, J. S., and Tomlin, C., "Inference Methods for Autonomous Stochastic Linear Hybrid Systems," *Hybrid Systems: Computation and Control*, Vol. 2993, edited by R. Alur and G. J. Pappas, Lecture Notes in Computer Science, Springer–Verlag, Philadelphia, March 2004, pp. 64–79.

[28] Shin, J., Lee, N., Thrun, S., and Guibas, L., "Lazy Inference on Object Identities in Wireless Sensor Networks," *Information Processing in Sensor Networks*, Los Angeles, April 2005.

[29] Sworder, D., and Boyd, J., *Estimation Problems in Hybrid Systems*, Cambridge Univ. Press, Cambridge, England, 1999.

[30] Sinkhorn, R., "Diagonal Equivalence to Matrices with Prescribed Row and Column Sums," *American Mathematical Monthly*, Vol. 74, 1967, pp. 402–405.

[31] Hwang, I., Balakrishnan, H., Roy, K., Shin, J., Guibas, L., and Tomlin, C., "Multiple-Target Tracking and Identity Management Algorithm for Air Traffic Control," *Proceedings of the Second IEEE International Conference on Sensors*, Toronto, Oct. 2003.

[32] Balakrishnan, H., Hwang, I., and Tomlin, C., "Polynomial Approximation Algorithms for Belief Matrix Maintenance in Identity Management," *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 2004.

[33] Li, X., and Bar-Shalom, Y., "Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking," *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 3, Sept. 1993, pp. 186–194.

[34] de Feo, M., Graziano, A., Migliolo, R., and Farina, A., "IMMJPDA Versus MHT and Kalman Filter with NN Correlation: Performance Comparison," *IEE Proceedings. Radar, Sonar, and Navigation*, Vol. 144, No. 2, April 1997, pp. 49–56.