Multiple-Target Tracking and Identity Management

Inseok Hwang[*], Hamsa Balakrishnan[*], Kaushik Roy[†], Jaewon Shin[†], Leonidas Guibas[‡] and Claire Tomlin[*†]

[*]Dept. of Aeronautics and Astronautics, Stanford University, CA 94305

[†] Electrical Engineering Dept., Stanford University, CA 94305

[‡] Computer Science Dept., Stanford University, CA 94305

{ishwang, hamsa, kroy1,jwshin,tomlin}@stanford.edu; guibas@cs.stanford.edu

Abstract

This paper involves the development of an algorithm which can simultaneously track and manage identities of multiple targets in a sensor network, for the purpose of air traffic control. We propose a logical integration of Joint Probabilistic Data Association (JPDA) [1], used for associating measurements with targets, and the Identity Management (IM) [2] algorithm for sensor networks, which utilizes target attribute information from local sensors to maintain the target's identity correctly. For target tracking, we use a modified version of the Interacting Multiple Model (IMM) algorithm called the Residual-Mean IMM (RMIMM) which we developed [3]. The proposed algorithm incorporates target state estimate information from the tracking algorithm into the evolution of a doubly-stochastic belief matrix for the target identities, and also assimilates any local information available. The algorithm has been shown not only to converge, but also to not increase the uncertainty in our belief.

INTRODUCTION

The multiple-target tracking problem deals with correctly tracking several targets given noisy sensor measurements at every instant, while the identity management problem tries to associate target identities with the state estimates available at every instant in a sensor network. Although closely related, the two problems have so far only been studied independently. Most multiple-target data association and tracking algorithms proposed until now do not attempt to use local attribute information about targets to improve their performance. While they emphasize the need to track several targets simultaneously, they do not address the necessity to distinguish between the different targets, and indeed, often lead to target-swapping while tracking. In practice, given a network of sensors, in addition to the continuous state measurements, we frequently also receive local sensor information about identities, which can be exploited to reduce target-swapping during tracking. In this paper, we develop an algorithm that solves these two related problems at the same time by efficiently incorporating signature information from local sensors into data association. The algorithm is a mathematically consistent way of combining the JPDA algorithm for data association, the IM algorithm for multiple-target identity management, and the RMIMM algorithm for target tracking.

Multiple-target Identity Management (MIM) algorithm[2]

A scalable distributed algorithm for computing and maintaining multi-target identity information has been developed using Identity-Mass Flow which overcomes the exponential computational complexity in managing multi-target identity. This algorithm maintains information about *who is who* over time given target position estimates. The main results are: the introduction of the *identity belief matrix*, a doubly stochastic matrix, that describes how the identity information of each target is represented, and the development of a distributed algorithm for computing and updating the identity belief matrix.

This algorithm assumes that the target position estimates are given, but in practice, it is difficult to get the target position state estimates accurately enough for the MIM algorithm to work in the multi-target tracking environment. Thus, data association and MIM are closely related since both algorithms compute and update the relation between tracks and targets, and the MIM can be used in the data association step of the multi-target tracking algorithm.

DATA ASSOCIATION ALGORITHMS

We consider the problems of associating measurements with targets and tracking one or more targets. A detailed description of this problem and the issues therein are presented in [1]. The key step in this process is the validation of measurements, it is also the reason why we believe this algorithm would apply easily to distributed systems.

Measurement validation: Denote $z(k)$ as a measurement at time k and $\hat{z}(k+1|k)$ as a predicted measurement at time $k+1$ using information up to time k. Assume $p[z(k+1)|Z^k] = \mathcal{N}[z(k+1)|\hat{z}(k|k+1), S(k+1)]$. Let there be T targets. The *validation region* (*or gate*) is defined as:

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

where $r(k+1) = z(k+1) - \hat{z}(k+1|k)$ is the residual, S is its covariance, and γ is a design parameter. If measurements that lie inside the gate are considered valid, the set of validated measurements at time k is given by:

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

where m_k is a random variable. In the case of multi-target tracking, if there is always a one to one mapping from targets to measurements, i.e., $\sum_k m_k = T$. The measurement sequence up to time k is defined as:

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

The problem of associating each validated measurement with an appropriate target is known as data association, measurement association, or data correlation. The set of validated measurements for a given target consists of both the potentially correct and incorrect measurements.

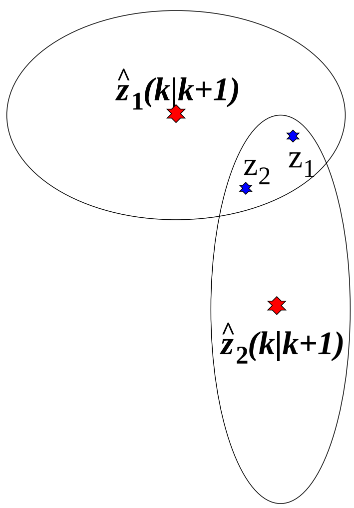

Figure 1. Validation Gate

For example, Figure 1 shows that measurements z_1 and z_2 are both validated for both target $\hat{z}_1$ and for target $\hat{z}_2$, giving two possible associations, only one of which is correct.

Nearest-Neighbor Standard Filter (NNSF)

The NNSF selects the validated measurement closest to the predicted measurement and uses it for state estimation. The distance measure is

$$d(z)^2 = r^T(k)S^{-1}r(k) \tag{4}$$

Since the filter-calculated covariance matrix S does not account for the possibility of processing incorrect measurement associations, the performance of the NNSF might be poor in some cases, resulting in the incorrect association of measurements to targets.

Joint Probabilistic Data Association Filter (JPDAF): a sub optimal Bayesian algorithm

If there are several targets in the same neighborhood, measurements from one target can fall in the validation gates of a neighboring target persistently. Thus the performance of a tracking algorithm could degrade significantly in such a situation, due to wrong association of measurement to target.

The key to the JPDAF algorithm is the evaluation of the conditional probabilities of the following joint events:

$$\Theta = \bigcap_{j=1}^{T} \theta_{jt_j}, \quad j = 1, \cdots, T; \;\; t = 0, 1, \cdots, T \tag{5}$$

where θ_{jt_j} := {measurement j originated from target t} and t_j is the index of the target to which measurement j is associated in the event under consideration. A joint event association matrix can be represented by the permutation matrix

$$\hat{\Omega} = [\hat{\omega}_{jt}(\Theta)], \;\; \text{where } \hat{\omega}_{jt}(\Theta) = \begin{cases} 1 & \text{if } \theta_{jt} \subset \Theta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We first define the following notation:

$$\begin{array}{l} \text{Target detection indicator:} \\ \delta_t(\Theta) := \sum_{j=1}^{m_k} \hat{\omega}_{jt}(\Theta) \leq 1, \\ \text{Measurement association indicator:} \\ \tau_j(\Theta) := \sum_{t=1}^{T} \hat{\omega}_{jt}(\Theta), \\ \text{Number of unassociated measurements in event } \Theta: \\ \phi(\Theta) := \sum_{j=1}^{m_k} [1 - \tau_j(\Theta)] \\ P_D : \text{target detection probability} \end{array} \tag{7}$$

Then, the marginal association probability is

$$\begin{array}{rcl} \beta_{jt} & = & \sum_{\Theta} P\{\Theta|Z^k\} \hat{\omega}_{jt}(\Theta) \\ P\{\Theta|Z^k\} & = & \frac{\phi!}{cV^{\phi}} \prod_{j=1}^{m_k} [\mathcal{N}_{t_j}(z_j(k))]^{\tau_j} \prod_{t=1}^{T} (P_D^t)^{\delta_t} (\bar{P}_D^t)^{1-\delta_t} \end{array} \tag{8}$$

where $\mathcal{N}_{t_j}[z_j(k)] := \mathcal{N}[z_j(k); \hat{z}^{t_j}(k|k-1), S^{t_j}(k)]$, $\bar{P}_D^t = 1 - P_D^t$, and $\hat{z}^{t_j}(k|k-1)$ denotes the predicted measurement for target t_j with an associated residual covariance S^{t_j}. In the case of a one-one association of targets to measurements (as in the case under study here), $\phi(\Theta) = 0$.

HYBRID STATE ESTIMATION ALGORITHM: RESIDUAL-MEAN INTERACTING MULTIPLE MODEL (RMIMM) ALGORITHM [3]

We use a hybrid estimation algorithm for state estimation for multiple-maneuvering-target tracking. In this section, we describe the general structure of the Interacting Multiple Model (IMM) algorithm and propose a modified IMM algorithm which uses information about the mean of the residual. We call this modified IMM algorithm the Residual-Mean IMM (RMIMM).

We consider a stochastic linear hybrid system with discrete-time, continuous-state dynamics:

$$\begin{array}{rcl} x(k+1) & = & A_j x(k) + w_j(k) \\ z(k) & = & C_j x(k) + v_j(k) \end{array} \tag{9}$$

and a Markov transition of the discrete state (mode) given by:

$$P[j(k+1)|i(k)] = H_{ij} \qquad i, j \in M = \{1, 2, \cdots, N\} \tag{10}$$

where $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$ are the state and the output respectively. M is the set of discrete states. The terms w and v are respectively the mode-dependent, uncorrelated, white Gaussian process noise and measurement noise with zero means and covariances Q_j and R_j. H_{ij} is the Markov mode transition probability from mode i to mode j. Thus, given the above system parameters, hybrid estimation is to estimate both the continuous state and the discrete state at time k from the measurement sequence up to time $k-1$ ($k = 1, 2, \cdots$). The

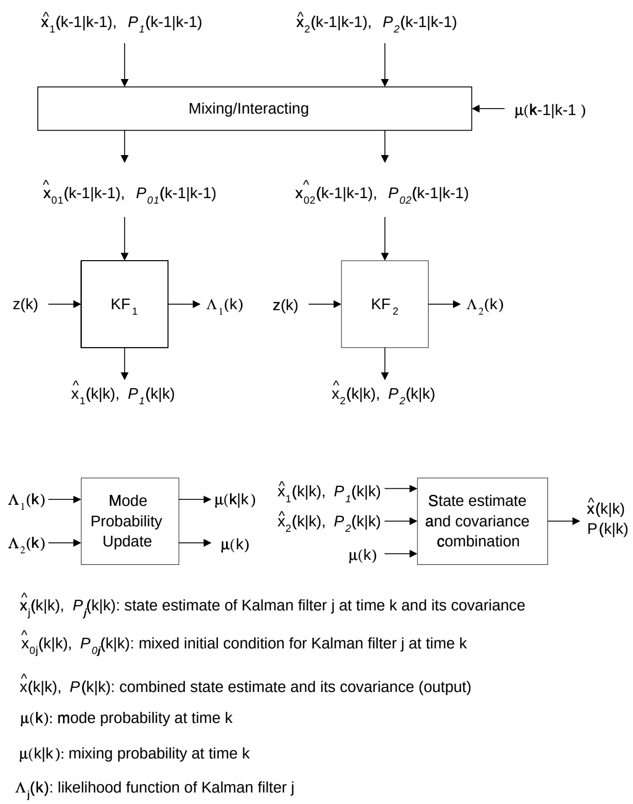

Figure 2. Structure of the IMM algorithm (for two modes) [4].

IMM algorithm is a multiple-model-based state estimation algorithm which computes the state estimate using a weighted sum of estimates from a bank of Kalman filters matched to different modes of the system. The general structure of the IMM algorithm as shown in Figure 2[4] is as follows:

Mixing probability: This is the probability that the system is in mode i at time $k-1$, given that it is in mode j at time k:

$$\mu_{ij}(k-1|k-1) = \frac{1}{c_j} H_{ij}\mu_i(k-1) \tag{11}$$

where c_j is a normalization constant, and where $\mu_i(k)$ is the mode probability of mode i at time k, i.e., a measure of how probable it is that the system 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.

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

$$\begin{aligned}\hat{x}_{0j}(k-1|k-1) &= \textstyle\sum_i \hat{x}_i(k-1|k-1)\mu_{ij}(k-1|k-1)\\ P_{0j}(k-1|k-1) &= \textstyle\sum_i \{P_i(k-1|k-1)\\ &\quad +[\hat{x}_i(k-1|k-1)\\ &\quad -\hat{x}_{0j}(k-1|k-1)][\hat{x}_i(k-1|k-1)\\ &\quad -\hat{x}_{0j}(k-1|k-1)]^T\}\mu_{ij}(k-1|k-1)\end{aligned} \tag{12}$$

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.

Kalman Filter: N Kalman filters run in parallel (multiple-model-based (hybrid) estimation).

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:

$$\Lambda_j(k) = \mathcal{N}(r_j(k); 0, S_j(k)) \tag{13}$$

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.

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

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\sum_i H_{ij}\mu_i(k-1) =: \frac{1}{c}\Lambda_j(k)\hat{\mu}_j(k) \tag{14}$$

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

Combination (output of the IMM): The state estimate is a weighted sum of the estimates from N Kalman filters and the mode estimate is the mode which has the highest mode probability:

$$\begin{aligned}\hat{x}(k|k) &= \sum_j \hat{x}_j(k|k)\mu_j(k)\\ P(k|k) &= \sum_j \{P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)]\\ &\qquad [\hat{x}_j(k|k) - \hat{x}(k|k)]^T\}\mu_j(k)\\ \hat{m}(k|k) &= \arg\max_j \mu_j(k)\end{aligned}$$

where $\hat{m}(k|k)$ is the mode estimate at time k.

As can be seen from the standard IMM algorithm, the mode probability in (14) depends on the likelihood function Λ_j. Thus, if the likelihoods of the modes are close to each other, the mode estimate may be inaccurate. Inaccurate mode estimates could produce poor state estimates, degrading the tracking accuracy. Because we are interested in using this for aircraft tracking, we propose a method which reduces false mode estimation by increasing the difference between the likelihood of the correct mode and the likelihoods of the other modes, using the fact that if the Kalman filter corresponding to mode j is the correct one, then the residual in (13) should be a white Gaussian process with a zero mean. Otherwise, its mean should not be zero. Therefore, we propose a new likelihood function:

$$\Lambda_j^{new}(k) = \begin{cases} \frac{N_j(k)\Lambda_j(k)}{\sum_{i=1}^{N} N_i(k)\Lambda_i(k)} & \text{if } \bar{r}_j(k) \neq 0 \\ \Lambda_j(k) & \text{otherwise} \end{cases} \quad (15)$$

where

$$N_i(k) = \begin{cases} \|\bar{r}_i(k)\|^{-1} & \text{if } \bar{r}_j(k) \neq 0 \\ 1 & \text{otherwise} \end{cases} \quad (16)$$

Proposition 1: The differences between the new likelihood function (15) for the correct mode and those for the incorrect mode, is greater than the corresponding differences using the previous likelihood function from (13).
Proof: See [3].
Thus, the RMIMM algorithm uses (15) instead of (13) as a mode likelihood.

MULTIPLE-TARGET TRACKING AND IDENTITY MANAGEMENT (MTIM) ALGORITHM

We consider multiple-target tracking and identity management problems in a no-clutter environment.

No-clutter environment

We assume that there is no clutter, i.e. there are T targets and T measurements at each time. Then, $P_D = 1$, $\phi(\Theta) = 0$, and $\tau_j(\Theta) = 1$. (8) becomes

$$\beta_{jt}(k) = \sum_{\Theta}[\prod_{j=1}^{T} N[z_j(k)]\hat{\omega}_{jt}(\Theta)] \quad (17)$$

After making the association matrix $A(k) := [\beta_{jt}(k)]$ a doubly-stochastic matrix $A'(k)$, we use $A'(k)$ as the mixing matrix $M(k)$ in the MIM. Then, the evolution of the belief matrix is

$$B(k) = B(k-1)M(k) \quad (18)$$

Entropy of Mixing Process

We define the entropy of the system as the average entropy over the distributions of our beliefs of the identities of the targets. In the belief matrix, since 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 rederive Lemma 1 from ([2]).
Lemma 1: Let $\bar{H}(B(k))$ be the average entropy over all the columns of the belief matrix $B(k)$, where the entropy of a column is the statistical entropy of its probability mass function. 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,

$$\begin{aligned} \bar{H}(B(k)) &= \frac{1}{n}\sum_{j=1}^{N} H(b_j(k)) & (19) \\ &= \frac{1}{n}\sum_{j=1}^{N} H([M(k)B(k-1)]_j) & (20) \\ &= \frac{1}{n}\sum_{j=1}^{N} H([\sum_{i=1}^{N!}\alpha_i\Phi_i B(k-1)]_j) & (21) \\ &= \frac{1}{n}\sum_{j=1}^{N} H(\sum_{i=1}^{N!}\alpha_i[\Phi_i B(k-1)]_j) & (22) \\ &\geq \frac{1}{n}\sum_{j=1}^{N}\sum_{i=1}^{N!}\alpha_i H([\Phi_i B(k-1)]_j) & (23) \\ & & (24) \end{aligned}$$

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

$$\Longrightarrow H([\Phi_i B(k-1)]_j) = H(b_j(k-1)) \quad (25)$$

Therefore, we get

$$\begin{aligned} \bar{H}(B(k)) &\geq \frac{1}{n}\sum_{j=1}^{N}\sum_{i=1}^{N!}\alpha_i H(b_j(k-1)) & (26) \\ &= \frac{1}{n}\sum_{j=1}^{N} H(b_j(k-1)), \text{since} \sum_{i=1}^{N!}\alpha_i = 1 & (27) \\ &= \bar{H}(B(k-1)) & (28) \end{aligned}$$

Corollary 1: Since $\bar{H}(B(k)) = \frac{1}{n}\sum_{j=1}^{N} H(b_j(k))$ (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 $B(k) = B(k-1)M(k)$.

Incorporation of Local Information

In the IM algorithm, we assume that local information arrives in the form of column updates to the corresponding columns. We then preserve that specific column and scale the rest of the belief matrix to make it doubly-stochastic, using Sinkhorn scaling.

Identity − type local information: This is local information which 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.

Conjecture 1: Identity-type local information always reduces the entropy of the system.

General forms of local information: In general, local information is in the form of a stochastic (elements sum to 1) column vector. In this case, clearly, the effect on the entropy depends on the elements of the column (related in some way to the relative entropies) and need not necessarily decrease the entropy. Consider, for example, the belief matrix $\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$. 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 entropy of the updated matrix is 0.6109, that is, the entropy increases when we incorporate information of this form.

The above 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. The Sinkhorn iteration is only an *approximate* algorithm to obtain a doubly-stochastic matrix, and there may be situations in which the incorporation of local information would increase our uncertainty in belief. However, we are justified in always incorporating identity-type information. Since 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.

Algorithm 1: Multiple-target Tracking and Identity Management algorithm

For target t, $t \in \{1, \cdots, T\}$,

- Step 1: mixing/interaction: $\hat{x}_{0i}(k-1|k-1)$ and $P_{01}(k-1|k-1)$
- Step 2: Kalman Filter i ($i \in \{1, \cdots, N\}$)
  - *State propagation/prediction*

$$\begin{aligned} \hat{x}_i(k|k-1) &= A_i \hat{x}_{0i}(k-1|k-1) \\ P_i(k|k-1) &= A_i P_{0i}(k-1|k-1) A_i^T + Q_i \\ S_i(k) &= C_i P_i(k|k-1) C_i^T + R_i \end{aligned} \tag{29}$$

  - *Measurement Validation*

$$r_{ij}^T S_i^{-1}(k) r_{ij}(k) < \gamma^2 \tag{30}$$

    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.
  - *Measurement Update*
    - Compute an association matrix using (17) and a mixing matrix.
    - Update the belief matrix using (18).
    - If local information arrives, then update the column corresponding to the local information, and scale the rest of the matrix (using Sinkhorn Scaling) to make it doubly-stochastic.
    - Update continuous state estimate and its covariance

$$\begin{aligned} \hat{x}_i(k|k) &= \hat{x}_i(k|k-1) \\ &\quad + K_i(k) \textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k) \\ P_i(k|k) &= [I - K_i(k) C_i] P_i(k-1|k-1) \\ &\quad + K_i(k) [\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k) r_{il}(k)^T \\ &\quad - (\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k)). \\ &\quad (\textstyle\sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k))^T] K_i(k)^T \end{aligned} \tag{31}$$

      where a Kalman filter gain is given by:

$$K_i(k) = P_i(k|k-1) C_i^T [C_i P(k|k-1) C_i^T + R_i]^{-1}. \tag{32}$$

- Step 3: Compute the mode likelihood functions

$$\Lambda_i(k) = \mathcal{N}(r_i(k); 0, S_i(k)) \tag{33}$$

  where $r_i(k) := \sum_{l=1}^{m_k^i} \beta_{il}(k) r_{il}(k)$.
- Step 4: Compute the mode probabilities: $\mu_i(k)$.
- Step 5: Compute the outputs: $\hat{x}(k|k)$, $P(k|k)$, and $B(k)$.

The outputs $\hat{x}(k|k)$ and $P(k|k)$ correspond to the hybrid state estimate (continuous state estimate and discrete mode estimate) of target t at time k. $B(k)$ corresponds to the the belief matrix, which reflects the probabilities of the identities of the different targets.

EXAMPLE: AIRCRAFT TRACKING EXAMPLE

In this section, we consider the tracking and identity management of multiple aircraft which interact with each other over a period of time. We consider a three-aircraft scenario, and the results are shown in Figures 3 and 4. We plot the estimated trajectories with the managed identities, and also provide the propagation of the belief matrix. In this example, local information of the identity-type is obtained at $t = 59$, as seen from the immediate improvement of the beliefs. It is evident from the figure that the algorithm we have proposed maintains an accurate estimate of the track as well as the identities in the presence of multiple possible associations of measurements to targets, provided there are sources of local information. Such a decrease in belief usually corresponds to interactions between targets, due to their proximity. This is seen in comparing the trajectories and belief matrix plots in Figures 3 and 4 respectively.

CONCLUSIONS

The results presented in this paper are a first step in the development of an algorithm that can simultaneously track and maintain identities of multiple targets. Although proposed for an Air Traffic Control scenario, we believe that these methods are easily extendable to the problem of tracking in other sensor networks. The Radar Surveillance System of the Air Traffic Control System can easily be replaced by any other network of sensors, for different applications. The algorithm exploits local interactions in both tracking (using validation gates) and identity-management (using the properties of Sinkhorn iterations), thus making it a promising candidate for tracking in distributed sensor networks. The same property of these algorithms also make them applicable to very large scale networks.

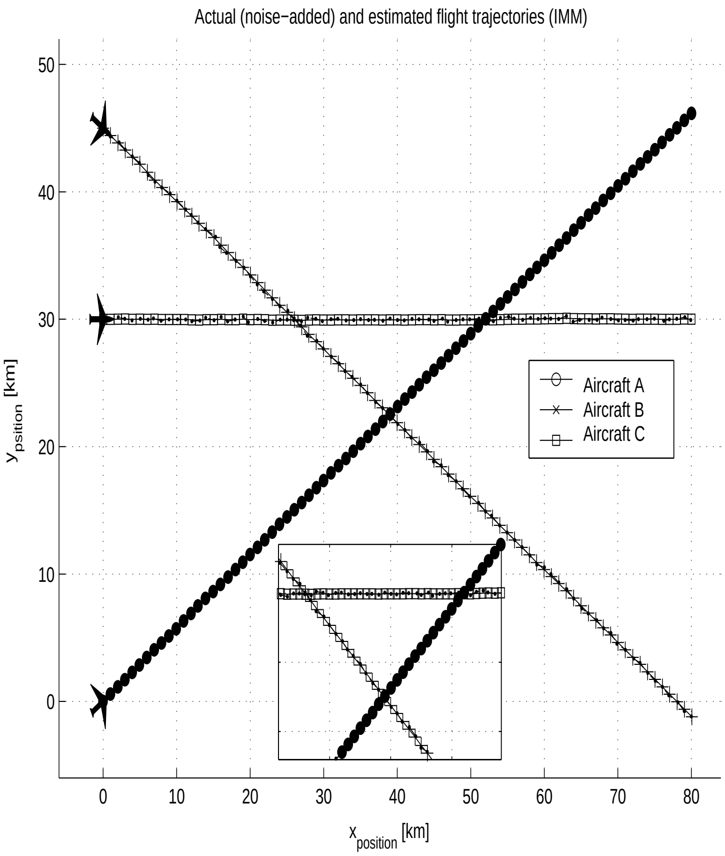

Figure 3. Tracking and identity management of three aircraft. Estimated trajectories and Identities (given by the legends)

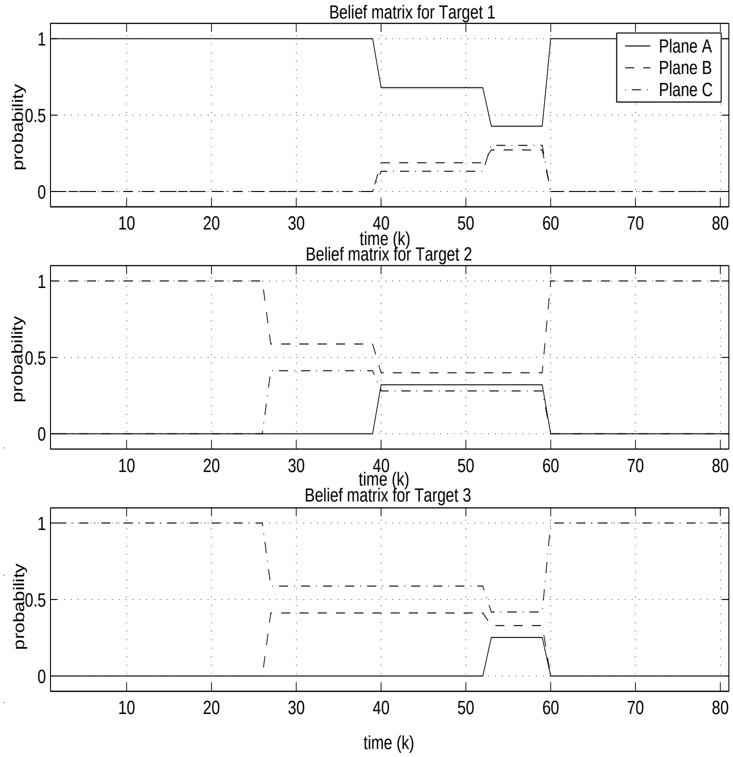

Figure 4. Tracking and identity management of three aircraft. Belief matrix propagation.

Current target classification and tracking algorithms depend on signal processing as a means of determining identity [5]. We believe that it might be power-efficient and computationally less expensive to use such signal-processing algorithms as a source of local information when necessary, and use algorithms such as the one we have proposed for tracking and identity management.

REFERENCES

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

[2] J. Shin, L.J. Guibas, and F. Zhao. A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. In F. Zhao and L. Guibas, editors, Information Processing in Sensor Networks, Lecture Notes in Computer Science 2654, pages 223–238, Palo Alto, CA, April 2003.

[3] I. Hwang, J. Hwang, and C. Tomlin. Flight-mode-based aircraft conflict detection using a Residual-Mean Interacting Multiple Model algorithm. In Proceedings of AIAA Guidance, Navigation, and Control Conference, Austin Texas, 2003. AIAA 2003-5340.

[4] X.R. Li and Y. Bar-Shalom. Design of an Interacting Multiple Model Algorithm for air traffic control tracking. IEEE Transactions on Control Systems Technology, 1(3):186–194, September 1993.

[5] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao. Target classification and localization in habitat monitoring. In Procedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Hong Kong, China, April 2003.