JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 17, No. 6, November–December 1994

Multisensor Multitarget Mixture Reduction Algorithms for Tracking

Lucy Y. Pao*
Northwestern University, Evanston, Illinois 60208

A single-sensor, single-target mixture reduction (MR) data association algorithm is extended for use in multisensor, multitarget tracking situations. MR is extended for tracking an arbitrary number of targets using an arbitrary number of sensors under the assumption that the sensor measurement errors are independent across sensors. Like the single-sensor, single-target MR algorithm, which gives better performance than the probabilistic data association (PDA) filter, the multisensor, multitarget MR extensions give similar improvements in performance compared to the joint PDA (JPDA) and multisensor JPDA (MSJPDA) algorithms. Further, in the formulations for the multisensor and multitarget MR algorithms, the equations for the calculation of the data association probabilities have been put in the same form as for the JPDA, thus allowing previously developed fast JPDA computational techniques to be applicable.

I. Introduction

IN the problem of tracking targets in random clutter, the optimal Bayesian solution leads to Gaussian mixture distributions that consist of an exponentially increasing number of components.[1] In practice, the number of components in each mixture is kept in check by approximating it with fewer components. The nearest neighbor (NN) approximation reduces each target's mixture to its largest component.[1–3] The joint probabilistic data association (JPDA) approximation merges the components of each target's mixture into a single Gaussian component.[1,4] However, when there are several significant, well-spaced components in the original mixture, important information may be lost by using the NN or JPDA methods. A more flexible approximation method for single-sensor, single-target tracking, known as mixture reduction (MR), has been proposed by Salmond.[5] This MR algorithm preserves the mean and covariance of the original mixture of components, successively merging components until the number of components is reduced to some user-specified limit.

In this paper, we first review Salmond's MR algorithm for single-sensor tracking of a single target in clutter. We then derive extensions to Salmond's MR algorithm for use in tracking multiple targets and when there are measurements available from multiple sensors. Just as the single-sensor, single-target MR algorithm gives better performance[5] than the probabilistic data association (PDA) filter, the multisensor, multitarget MR extensions are shown to give similar improvements in performance as compared to the JPDA and multisensor JPDA (MSJPDA)[6] algorithms. Extension of the MR algorithm for tracking of multiple targets is necessary for use in realistic tracking scenarios where there are usually many targets that need to be tracked. Moreover, many current tracking systems have been designed with multiple sensors, thus requiring the further extension to a multisensor algorithm.

The paper is organized as follows. In Sec. II, the target dynamics and measurement equations are defined. We review the single-sensor, single-target MR algorithm[5] in Sec. III, and we extend the algorithm to the multitarget and multisensor cases in Sec. IV and V, respectively. In Sec. VI, simulation results evaluating the performances of the MR extensions are presented. Finally, concluding remarks are given in Sec. VII.

Received June 1, 1993; revision received Feb. 14, 1994; accepted for publication Feb. 22, 1994.

*Assistant Professor, Electrical Engineering and Computer Science Department. Member AIAA.

II. Multisensor, Multitarget Tracking

The multisensor, multitarget tracking problem is to track T targets in clutter with N_s sensors. Measurements (also called reports or returns) from the sensors are received by the processor at discrete time intervals. Each measurement can originate from at most one target. Some sensors may not provide measurements at every interval. Some of the measurements arise from targets, and some arise from clutter; some targets may not yield any measurements at all in a particular time interval or for a particular sensor. Measurement errors due to measurements from one sensor are assumed to be independent of those from another sensor.

Let $\boldsymbol{x}^t(k)$ $(1 \le t \le T)$ denote the state vectors of each target t at the kth time interval. Suppose the target dynamics are determined by known matrices $\boldsymbol{F}^t(k)$ and $\boldsymbol{G}^t(k)$ and random noise vectors $\boldsymbol{w}^t(k)$ as follows:

$$\boldsymbol{x}^t(k+1) = \boldsymbol{F}^t(k)\boldsymbol{x}^t(k) + \boldsymbol{G}^t(k)\boldsymbol{w}^t(k) \tag{1}$$

where $t = 1, \ldots, T$. The noise vectors $\boldsymbol{w}^t(k)$ are stochastically independent Gaussian random variables with zero mean and known covariance matrices.

With N_s sensors, let $M_{i,k}, i = 1, 2, \ldots, N_s$ be the number of reports from each sensor i at the kth time interval. Assuming a precorrelation gating process is used to eliminate some of the returns,[1] let $m_{i,k}$ denote the number of validated returns from sensor i at time k. The target-originated measurements, if there are any, are determined by

$$\boldsymbol{z}_{i,l}^t(k) = \boldsymbol{H}_i(k)\boldsymbol{x}^t(k) + \boldsymbol{v}_i^t(k) \tag{2}$$

where $t = 1, \ldots, T$, $i = 1, \ldots, N_s$, and $1 \le l \le m_{i,k}$. The $\boldsymbol{H}_i(k)$ matrices are known, each $\boldsymbol{v}_i^t(k)$ is a zero-mean Gaussian noise vector uncorrelated with all other noise vectors, and the covariance matrices of the noise vectors $\boldsymbol{v}_i^t(k)$ are known. For a given target t and sensor i, it is not known which measurement l $(1 \le l \le m_{i,k})$ originates from the target. That is the problem of data association whereby it is necessary to determine which measurements originate from which targets.[1–3] In any time interval, it is assumed that a target can give rise to at most one measurement from a particular sensor. Measurements not originating from targets are known as false measurements (i.e., clutter), and we assume that false measurements are uniformly distributed throughout the surveillance region. Let $\mathcal{Z}(k)$ denote the observation at time k:

$$\mathcal{Z}(k) = \big(z_{1,1}(k), \ldots, z_{1,m_{1,k}}(k),$$

$$z_{2,1}(k), \ldots, z_{2,m_{2,k}}(k), \ldots, z_{N_s,1}(k), \ldots, z_{N_s,m_{N_s,k}}(k)\big)$$

The t superscripts are not indicated, since we do not know which measurements originate from which target. Let $\mathcal{Z}^k$ denote the

sequence of the first k observations; that is, $\mathcal{Z}^k = (\mathcal{Z}(1), \ldots, \mathcal{Z}(k))$. The goal of tracking algorithms is, at each time interval, to associate measurements to targets based on the current (and sometimes past) estimates of the target states and then to update these estimates. In the following sections, we will omit the dependence on k where it is clear from the context.

III. Single-Sensor, Single-Target MR

In this section, we review a MR algorithm[5] for single-sensor, single-target tracking that approximates a mixture of components (or modes) representing a target by successively merging components until the number of components is reduced to less than or equal to some user-specified limit, N_T. For single-sensor, single-target tracking, the target dynamics and measurements are governed by Eqs. (1) and (2) with $T = 1$ and $N_s = 1$. Suppose there are $n_1 \leq N_T$ components after the MR iterations are complete at time $k-1$, each component having a probability weighting ψ^j, $j = 1, \ldots, n_1$, associated with it. At time step k, another set of measurements is received, and there are $m_{1,k}$ validated measurements after gating. Then, there may be as many as $N = n_1(m_{1,k}+1)$ components due to the new measurements, depending upon how many measurements gate with each mode of the target.

The probability weightings on the new modes, χ_l^j (new mode due to measurement l and previous mode j), are computed in a similar way as with the optimal Bayesian filter for single-sensor, single-target tracking.[1] Define the likelihood

$$p_l^j = \begin{cases} \psi^j \lambda (1 - P_d), & l = 0 \\ \psi^j N_j(z_l(k)) P_d, & l \neq 0, \quad \omega_l^j = 1 \\ 0, & l \neq 0, \quad \omega_l^j = 0 \end{cases} \tag{3}$$

where λ is the clutter density, P_d is the probability of detection of the target, $N_j(\cdot)$ is the normal distribution with mean and covariance for the target-originated measurement as computed by the Kalman filter for the jth mode of the target, and ω_l^j is a binary function indicating whether return l gated with mode j of the target. The return number $l = 0$ indicates the possibility of no return from the sensor being associated with the target. Then, for an appropriate normalization constant c, the probabilities for the new modes are

$$\chi_l^j \triangleq \frac{1}{c} p_l^j \tag{4}$$

Because the modes represent exclusive and exhaustive hypotheses, all the χ_l^j sum to unity:

$$\sum_{j=1}^{n_1} \sum_{l=0}^{m_{1,k}} \chi_l^j = 1 \tag{5}$$

Thus, the χ_l^j can be found by determining the products $c\chi_l^j$ from Eqs. (3) and (4), solving for c using Eq. (5), and normalizing. Renaming the χ_l^j as β_i, where $i = n_1 l + j$, we obtain a set of $n_1(m_{1,k}+1)$ number of β_i (some of which are identically 0). The corresponding state estimates and covariances $\hat{\boldsymbol{x}}_i$ and $\boldsymbol{P}_i$ are obtained from the Kalman filtering equations using the appropriate measurement z_l and target mode state prediction $\hat{\boldsymbol{x}}^j$.

Thus, there is a mixture distribution for the target state vector of the form

$$p(\boldsymbol{x}) = \sum_{i=1}^{N} \beta_i p_i(\boldsymbol{x}) \tag{6}$$

where $\boldsymbol{x}$ is the state vector of the target, N is the number of components in the mixture, $p_i(\boldsymbol{x})$ is a component probability density function (pdf), and β_i is a probability associated with the ith component such that $\beta_i > 0$ and $\sum_{i=1}^{N} \beta_i = 1$ [as in Eq. (5)].

Before new measurements are received at the next time interval, the β_i, $\hat{\boldsymbol{x}}_i$, and $\boldsymbol{P}_i$ are merged by the MR procedure. The algorithm is based on the proposition that the mixture components with the largest probabilities carry the most important information. Hence, the component corresponding to the largest probability β_i is called the principal component and components that are "close" to the principal component are clustered together with the principal component. The distance measure chosen to represent the closeness of components to the principal component is defined by

$$D_i^2 = \frac{\beta_i \beta_c}{\beta_i + \beta_c} (\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_c)^T \boldsymbol{P}_c^{-1} (\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_c) \tag{7}$$

where β_c, $\hat{\boldsymbol{x}}_c$, and $\boldsymbol{P}_c$ are the probability, mean, and covariance of the principal component and β_i and $\hat{\boldsymbol{x}}_i$ are the probability and mean of the ith component. This is a squared-distance measure with the covariance of the principal component as a normalization factor and with another scaling factor, $\beta_i \beta_c / (\beta_i + \beta_c)$. The $\beta_i \beta_c / (\beta_i + \beta_c)$ factor biases the distance measure so that small components are more easily clustered whereas large components retain their individuality. Any component i for which $D_i^2 < T_1$ is selected as a cluster member. The clustering boundary is a hyperellipsoid in state space that is a surface of constant probability density of the principal component. The threshold T_1 defines the acceptable modification to the distribution.

The cluster of components (some clusters may consist of a single component) is approximated by a single Gaussian defined by

$$\psi^j = \sum_{i \in \mathcal{C}} \beta_i \tag{8}$$

$$\hat{\boldsymbol{x}}^j = \frac{1}{\psi^j} \sum_{i \in \mathcal{C}} \beta_i \hat{\boldsymbol{x}}_i \tag{9}$$

$$\boldsymbol{P}^j = \frac{1}{\psi^j} \sum_{i \in \mathcal{C}} \beta_i \left(\boldsymbol{P}_i + \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i^T \right) - \hat{\boldsymbol{x}}^j (\hat{\boldsymbol{x}}^j)^T \tag{10}$$

where the set $\mathcal{C}$ contains the indices of all the components in the cluster and the probability, mean, and covariance of the cluster approximation are denoted by ψ^j, $\hat{\boldsymbol{x}}^j$, and $\boldsymbol{P}^j$, respectively. The index $j = 1$ indicates the approximation of the first cluster of components in the original mixture. With the approximation of Eqs. (8–10), the overall mean $\hat{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$ of the original mixture are the same as the overall mean and covariance of the resulting approximation mixture.[5,7]

This clustering process is iterated on the remaining components of the original mixture, incrementing j each iteration. After all components of the original mixture have been clustered, the approximation is complete if the number of components in the approximation is at or below the user-specified maximum N_T. Otherwise the threshold T_1 is increased by ΔT_1 and the clustering procedure is repeated (resetting j to 1) on the first approximation. This clustering operation is iterated until the necessary reduction has been effected. Target predictions are made based upon the final ψ^j, $\hat{\boldsymbol{x}}^j$, and $\boldsymbol{P}^j$ of the mixture approximation. At the next time step, new measurements are received and gated, and the entire procedure is repeated.

The choice of T_1 and ΔT_1 is a compromise between the number of iterations required and the possibility of clustering more components than necessary. Suggested threshold values[5,7] are $T_1 = 0.05 T_1'$ and $\Delta T_1 = 0.05 \Delta T_1'$, where T_1' and $T_1' + \Delta T_1'$ define hyperellipsoids containing 1 and 6%, respectively, of the probability mass of the principal component. The values of T_1' and $\Delta T_1'$ can be found from tables of chi-squared probabilities with the number of degrees of freedom equal to the dimension of the target state vectors $\boldsymbol{x}^t$.

For $N_T = 1$, the MR algorithm reduces to the PDA algorithm.[1] When $N_T > 1$, the MR algorithm allows a better approximation of the mixture of Gaussian components than the PDA algorithm. For a range of normalized clutter densities and normalized acceleration variances, Monte Carlo simulations of the MR algorithm with $N_T = 20$ have been shown[5,7] to yield "times to track loss" three to seven times longer than for the PDA algorithm. For a nominal normalized clutter density and a nominal normalized acceleration variance, the average times to track loss were about seven times longer than for the PDA for all values of $N_T \geq 10$. The average CPU time per time interval for these cases was found to be about eight times longer for the MR than for the PDA algorithm. Thus, the MR algorithm gives better performance (according to the time-to-track-loss metric) with an increase in computational complexity.

IV. Extension to Multiple Targets

In this section, we extend the single-sensor, single-target MR algorithm to a single-sensor, multiple-target MR (MTMR) algorithm. When there are multiple targets, the reduction of components for each target to less than or equal to a user-specified number N_T (assumed the same for all targets) is done as described above for a single target. However, when new measurements are received, the χ probability weightings on the new modes are computed differently than in Eqs. (3–5). The extension made here is analogous to the extension of the PDA algorithm to the JPDA algorithm.[1] Since a measurement for one target may fall into the gate of another target, this interference caused by another target must be treated differently than random clutter. In this section, we derive the equations for computing the weightings on the components after receiving a new set of measurements. For each target, the number of components is then reduced to less than or equal to the predetermined number N_T using the reduction algorithm described in the previous section.

The target states and measurements are governed by Eqs. (1) and (2) with $N_s = 1$. At time interval k, there are $m_{1,k}$ validated returns, and each target t ($1 \le t \le T$) has $n_t \le N_T$ modes. Define the mapping a: $\{1, \ldots, T\} \to \{0, 1, \ldots, m_{1,k}\}$ as a mapping that associates with each target t a return number $a(t)$, with the return 0 denoting the possibility of no return from the sensor for a target. For each association a, let $\Theta_a(k)$ denote the event that a is the correct association for the kth observation. For $1 \le t \le T$ and $0 \le l \le m_{1,k}$, let $\theta_l^t(k)$ denote the event that $a(t) = l$ for the correct association $a(t)$ for the kth observation. The event $\theta_l^t(k)$ is a disjoint union of events $\Theta_a(k)$:

$$\theta_l^t(k) = \biguplus_{a:\ a(t)=l} \Theta_a(k) \tag{11}$$

where a: $a(t) = l$ indicates all mappings a in which report l originates from target t. Let $\beta_l^t(k)$ denote the conditional probability of the event $\theta_l^t(k)$ given Z^k.

When there are multiple modes for each target, for each mapping a, there exists a set of mappings

$$a_{mr}\colon \{1_1, \ldots, 1_{n_1}, \ldots, T_1, \ldots, T_{n_T}\} \to \{0, 1, \ldots, m_{1,k}\}$$

that associate with each mode j of each target t the return number $a_{mr}(t, j)$. For each mapping a, there may be up to

$$N_{mr} = \prod_{t=1}^{T} n_t \tag{12}$$

corresponding a_{mr} mappings depending upon whether each return in the mapping a gates with all modes of its associated target. For each association a_{mr}, let $\Theta_{a_{mr}}(k)$ denote the event that a_{mr} is the correct association for the kth observation. For $1 \le t \le T$, $1 \le j \le n_t$, and $0 \le l \le m_{1,k}$, let $\theta_l^{t,j}(k)$ denote the event that $a_{mr}(t, j) = l$ for the correct association $a_{mr}(t, j)$ for the kth observation.

Let us define the following indicator function that depends on the mapping a for the simple event Θ_a:

$$\sigma_l(a) \triangleq \begin{cases} 0, & a(t) = 0 \\ 1, & \text{otherwise} \end{cases} \tag{13}$$

Further, let us define the following indicator functions that depend on each mapping a_{mr} for the simple event $\Theta_{a_{mr}}$:

$$\sigma_{t,j}(a_{mr}) \triangleq \begin{cases} 0, & a_{mr}(t, j) = 0 \\ 1, & \text{otherwise} \end{cases} \tag{14}$$

$$\tau_l(a_{mr}) \triangleq \begin{cases} 0, & a_{mr}^{-1}(l) = 0 \\ 1, & \text{otherwise} \end{cases} \tag{15}$$

where

$$a_{mr}^{-1}(l) \triangleq \begin{cases} (t, j), & a_{mr}(t, j) = l \\ 0, & a_{mr}(t, j) \ne l \end{cases} \qquad \forall t, j \tag{16}$$

The number of false reports in the joint event $\Theta_{a_{mr}}$ is

$$\phi(a_{mr}) \triangleq \sum_{l=1}^{m_{1,k}} [1 - \tau_l(a_{mr})] \tag{17}$$

Let $\chi_l^{t,j}(k)$ denote the conditional probability of the event $\theta_l^{t,j}$ given Z^k. Since $\theta_l^{t,j}$ is a disjoint union of events $\Theta_{a_{mr}}$, we have

$$\chi_l^{t,j} = \sum_{a_{mr}:\ a_{mr}(t,j)=l} P\left(\Theta_{a_{mr}} \mid Z^k\right) \tag{18}$$

Rewriting $P(\Theta_{a_{mr}} \mid Z^k)$ using Bayes's rule gives us

$$\begin{aligned} P\left(\Theta_{a_{mr}} \mid Z^k\right) &= P\left(\Theta_{a_{mr}} \mid Z(k), Z^{k-1}\right) \\ &= \frac{1}{c} f\left(Z(k) \mid \Theta_{a_{mr}}, Z^{k-1}\right) P\left(\Theta_{a_{mr}}\right) \end{aligned} \tag{19}$$

where c is an appropriate normalization constant. The density function on the right-hand side is

$$f\left(Z(k) \mid \Theta_{a_{mr}}, Z^{k-1}\right) = \prod_{l=1}^{m_{1,k}} f\left(z_l(k) \mid \theta_l^{a_{mr}^{-1}(l)}, Z^{k-1}\right) \tag{20}$$

by independence. The PDF on the right-hand side of Eq. (20) is

$$f\left(z_l(k) \mid \theta_l^{a_{mr}^{-1}(l)}, Z^{k-1}\right) = \begin{cases} \psi^{a_{mr}^{-1}(l)} N_{a_{mr}^{-1}(l)}(z_l(k)), & \tau_l(a_{mr}) = 1 \\ V_k^{-1}, & \tau_l(a_{mr}) = 0 \end{cases} \tag{21}$$

where V_k is the volume of the surveillance region at time k. Combining Eqs. (20) and (21) gives

$$f\left(Z(k) \mid \Theta_{a_{mr}}, Z^{k-1}\right) = V_k^{-\phi(a_{mr})} \times \prod_{t=1}^{T} \prod_{j=1}^{n_t} \left\{\psi^{t,j} N_{t,j}\left(z_{a_{mr}(t,j)}\right)\right\}^{\sigma_{t,j}(a_{mr})} \tag{22}$$

We now compute $P(\Theta_{a_{mr}})$, the last factor in Eq. (19). Denote by $\sigma(a_{mr})$ the vector of target mode detection indicators (14) corresponding to event $\Theta_{a_{mr}}(k)$. Note that given $\Theta_{a_{mr}}(k)$, the vector $\sigma(a_{mr})$ and the number $\phi(a_{mr})$ of false returns are determined. Hence,

$$\begin{aligned} P\left(\Theta_{a_{mr}}\right) &= P\left(\Theta_{a_{mr}}, \sigma(a_{mr}), \phi(a_{mr})\right) \\ &= P\left(\Theta_{a_{mr}} \mid \sigma(a_{mr}), \phi(a_{mr})\right) P\left(\sigma(a_{mr}), \phi(a_{mr})\right) \end{aligned} \tag{23}$$

The first factor is obtained by observing that in event $\Theta_{a_{mr}}$, the set of targets assumed detected contains $m_{1,k} - \phi(a_{mr})$ targets. Since each target can give rise to at most one measurement from a sensor, in this single-sensor case, each measurement can only be assigned to one mode of one target in each event $\Theta_{a_{mr}}$. Thus, the number of measurement-to-target-mode assignment events $\Theta_{a_{mr}}$ in which the same set of target modes is detected is given by the number of permutations of the $m_{1,k}$ measurements taken as $m_{1,k} - \phi(a_{mr})$, the number of targets to which a measurement is assigned under the same detection event. Hence, assuming each such event is equally likely, we have that

$$P\left(\Theta_{a_{mr}} \mid \sigma(a_{mr}), \phi(a_{mr})\right) = \frac{\phi!}{m_{1,k}!} \tag{24}$$

The other factor in Eq. (23) is

$$P\left(\sigma(a_{mr}), \phi(a_{mr})\right) = \frac{1}{N_{mr}} \prod_{t=1}^{T} \left(P_d^t\right)^{\sigma_t} \left(1 - P_d^t\right)^{1-\sigma_t} \mu_F(\phi) \tag{25}$$

where P_d^t is the probability of detection of target t and $\mu_F(\phi)$ is the probability of the number of false measurements. The σ_t indicators have been used in Eq. (25) to select the probabilities of detection

and no detection events in the event Θ_a corresponding to the event $\Theta_{a_{mr}}$ under consideration. The $1/N_{mr}$ factor, where N_{mr} is defined in Eq. (12), normalizes for the number of target mode detection vectors $\sigma(a_{mr})$ having the same set of target detection indicators $\sigma_t(a)$. Assuming a Poisson distribution for the clutter model,

$$\mu_F(\phi) = e^{-\lambda V}\frac{(\lambda V)^{\phi}}{\phi!} \tag{26}$$

Combining Eqs. (24–26) into Eq. (23) yields the prior probability of the joint association event $\Theta_{a_{mr}}$:

$$P\left(\Theta_{a_{mr}}\right) = \frac{e^{-\lambda V}}{N_{mr}}\frac{(\lambda V)^{\phi}}{m_{1,k}!}\prod_{t=1}^{T}\left(P_d^t\right)^{\sigma_t}\left(1-P_d^t\right)^{1-\sigma_t} \tag{27}$$

Similarly, combining Eqs. (22) and (27) into Eq. (19) yields the posterior probability of $\Theta_{a_{mr}}$:

$$P\left(\Theta_{a_{mr}} \mid \mathcal{Z}^k\right) = \frac{1}{c}\frac{e^{-\lambda V}}{N_{mr}}\frac{\lambda^{\phi}}{m_{1,k}!}\prod_{t=1}^{T}\left(P_d^t\right)^{\sigma_t}\left(1-P_d^t\right)^{1-\sigma_t}$$

$$\times\prod_{j=1}^{n_t}\left[\psi^{t,j}N_{t,j}\left(z_{a_{mr}(t,j)}\right)\right]^{\sigma_{t,j}} \tag{28}$$

In the product over the target modes, at most one factor will be nonunity since at most one report can be associated with any target.

The event probability $\beta_l^t(k)$ is then computed as

$$\begin{aligned}\beta_l^t &= P\left(\theta_l^t \mid \mathcal{Z}^k\right)\\ &= \sum_{a:\ a(t)=l} P\left(\Theta_a \mid \mathcal{Z}^k\right)\\ &= \sum_{a_{mr}:\ a_{mr}(t,\cdot)=l} P\left(\Theta_{a_{mr}} \mid \mathcal{Z}^k\right)\\ &= \sum_{j=1}^{n_t}\chi_l^{t,j}\\ &= \sum_{j=1}^{n_t}\sum_{a_{mr}:\ a_{mr}(t,j)=l}\frac{\lambda^{m_{1,k}-T}}{c'}\prod_{u=1}^{T}\left(P_d^u\right)^{\sigma_u}\left[\lambda\left(1-P_d^u\right)\right]^{1-\sigma_u}\\ &\quad\times\prod_{i=1}^{n_t}\left[\psi^{u,i}N_{u,i}\left(z_{a_{mr}(u,i)}\right)\right]^{\sigma_{u,i}}\end{aligned} \tag{29}$$

where c' is a normalization constant. The double sum in the final equation sums over the set of mappings a_{mr}: $a_{mr}(t, j) = l$ that constitute the mappings a: $a(t) = l$. Thus, Eq. (29) can be rewritten as

$$\beta_l^t = \sum_{a:\ a(t)=l}\sum_{j=1}^{n_t}\frac{\lambda^{m_{1,k}-T}}{c'}\prod_{u=1}^{T}\left(P_d^u\right)^{\sigma_u}$$

$$\times\left[\lambda\left(1-P_d^u\right)\right]^{1-\sigma_u}\omega_{a(u)}^{u,j}\psi^{u,j}N_{u,j}\left(z_{a(u)}\right) \tag{30}$$

where $\omega_{a(u)}^{u,j}$ is an indicator function denoting whether measurement $a(u)$ lies in the validation gate of mode j of target u; this $\omega_{a(u)}^{u,j}$ factor allows terms to be included in the sums only when return $a(u)$ falls in the validation gate of mode j of target u. Equation (30) can be rewritten as

$$\beta_l^t = \sum_{a:\ a(t)=l}\frac{\lambda^{m_{1,k}-T}}{c'}\prod_{u=1}^{T}\left(P_d^u\right)^{\sigma_u}\left[\lambda\left(1-P_d^u\right)\right]^{1-\sigma_u}$$

$$\times\sum_{j=1}^{n_t}\omega_{a(u)}^{u,j}\psi^{u,j}N_{u,j}\left(z_{a(u)}\right) \tag{31}$$

The association probability β_l^t is then given by

$$\beta_l^t = \frac{1}{c_n}\sum_{a:\ a(t)=l}\prod_{u=1}^{T}p_{a(u)}^u \tag{32}$$

where c_n is a normalization constant (for all β_l^t, i.e., for all t and l) and

$$p_l^t = \begin{cases}\lambda\left(1-P_d^t\right), & l=0\\ P_d^t\displaystyle\sum_{j=1}^{n_t}\omega_l^{t,j}\psi^{t,j}N_{t,j}(z_l), & \text{otherwise}\end{cases} \tag{33}$$

Since the events θ_l^t are mutually exclusive and exhaustive over l (i.e., either a return originated from target t or no return originated from target t), their probabilities sum to unity for each t:

$$\sum_{l=0}^{m_k}\beta_l^t = 1 \tag{34}$$

Hence, β_l^t can be found by determining the products $c_n\beta_l^t$ from Eqs. (32) and (33), solving for c_n using Eq. (34), and normalizing. The computation of the β_l^t association probabilities has been formulated to yield association probability equations that have a similar structure as for the JPDA filter. Thus, the computation of the β_l^t probabilities can take advantage of fast JPDA algorithms that have been developed.[8]

Now the $\chi_l^{t,j}$ modal association probabilities can be easily computed from the β_l^t association probabilities as

$$\chi_l^{t,j} = \begin{cases}0, & p_l^t = 0\\ \dfrac{p_l^{t,j}}{p_l^t}\beta_l^t, & \text{otherwise}\end{cases} \tag{35}$$

where

$$p_l^{t,j} = \begin{cases}\psi^{t,j}\lambda\left(1-P_d^t\right), & l=0 \quad \forall\, j=1,\ldots,n_t\\ P_d^t\psi^{t,j}N_{t,j}(z_l), & l\neq 0, \quad \omega_l^{t,j}=1\\ 0, & l\neq 0, \quad \omega_l^{t,j}=0\end{cases} \tag{36}$$

Equations (35) and (36) just state that β_l^t is distributed among the modes of target t.

For each target t, there are $n_t \leq N_T$ $(1 \leq t \leq T)$ modes prior to the receipt of new measurements. With the $m_{1,k}$ new validated measurements, the number of modes for each target t grows to as many as $n_t(m_{1,k}+1)$. The state and covariance estimates $\hat{\boldsymbol{x}}_l^{t,j}$ and $\boldsymbol{P}_l^{t,j}$ of each of the new modes are obtained by the Kalman filtering equations using the appropriate measurement z_l and target mode state prediction $\hat{\boldsymbol{x}}^{t,j}$. The probability weightings $\chi_l^{t,j}$ for the new modes are computed by Eqs. (35) and (36). Before the receipt of measurements at the next time interval, for each target t, the mixture of $\chi_l^{t,j}$, $\hat{\boldsymbol{x}}_l^{t,j}$, and $\boldsymbol{P}^{t,j}$ is approximated with $n_t \leq N_T$ components using the MR procedure described in Sec. III. Predictions for the next time interval are made based upon this approximation; and at the next time step, new measurements are received and the entire procedure is repeated.

V. Extension to Multiple Sensors

In this section, we extend the MTMR algorithm to the case where multiple sensors are used, where we assume that measurement errors are independent across sensors. We call this multisensor algorithm a multisensor, multitarget mixture reduction (MSMTMR) algorithm. The extension made here is similar to that of generalizing the multitarget JPDA algorithm for use with multisensor measurements.[6]

For multisensor, multitarget tracking, the target dynamics and measurements are governed by Eqs. (1) and (2). As with the MTMR algorithm, the number of modes for each target t is reduced to $n_t \leq N_T$ modes at each time step. Let

$$a_{ms}\colon \{1,\ldots,T\} \to \left\{\{0,1,\ldots,m_{1,k}\},\right.$$

$$\left.\{0,1,\ldots,m_{2,k}\},\ldots,\{0,1,\ldots,m_{N_s,k}\}\right\}$$

be a mapping at time k that associates each target t with a set of N_s numbers (one for each sensor) $a_{ms}(t)$ of returns. For each mapping a_{ms}, there is a set of N_s mappings a_i: $\{1,\ldots,T\} \to \{0,1,\ldots,m_{i,k}\}$ that associates return $a_i(t)$ from sensor i with each target t.

For each association a_{ms}, let $\Theta_{a_{ms}}(k)$ and each $\Theta_{a_i}(k)$ denote the events that a_{ms} and a_i, respectively, are the correct associations for the kth observation. For $1 \le t \le T$ and $\mathcal{L} = (l_1, l_2, \ldots, l_{N_s})$ where $0 \le l_1 \le m_{1,k}, \ldots, 0 \le l_{N_s} \le m_{N_s,k}$, let $\theta^t_{\mathcal{L}}(k)$ denote the event that $a_{ms}(t) = \mathcal{L}$ for the correct association a_{ms} for the kth observation. For $1 \le t \le T$, $1 \le i \le N_s$, and $0 \le l \le m_{i,k}$, let $\theta^t_{l,i}(k)$ indicate the event that $a_i(t) = l$ for the correct association a_i of sensor i for the kth observation. The event $\theta^t_{\mathcal{L}}(k)$ is a disjoint union of events $\Theta_{a_{ms}}(k)$, and $\theta^t_{l,i}(k)$ is a disjoint union of events $\Theta_{a_i}(k)$. Let $\beta^t_{\mathcal{L}}(k)$ denote the conditional probability of $\theta^t_{\mathcal{L}}$ given $\mathcal{Z}^k$, and let $\beta^t_{l,i}(k)$ denote the conditional probability of the event $\theta^t_{l,i}$ given $\mathcal{Z}^k$. The probability $\beta^t_{l,i}(k) = P(\theta^t_{l,i}(k) \mid \mathcal{Z}^k)$ is just the single-sensor association probability that return l of sensor i is associated with target t, and it can be calculated using Eqs. (32–34) derived in Sec. IV.

With each target t having $n_t \le N_T$ modes, let

$$a_{mr_{ms}}\colon \{1_1, \ldots, 1_{n_1}, \ldots, T_1, \ldots, T_{n_T}\} \to \{\{0, 1, \ldots, m_{1,k}\}, \{0, 1, \ldots, m_{2,k}\}, \ldots, \{0, 1, \ldots, m_{N_s,k}\}\}$$

be the mapping that associates each mode j of each target t with a set of N_s numbers (one for each sensor) $a_{mr_{ms}}(t, j)$ of returns. For each mapping $a_{mr_{ms}}(t, j)$, there is a set of N_s mappings

$$a_{mr_i}\colon \{1_1, \ldots, 1_{n_1}, \ldots, T_1, \ldots, T_{n_T}\} \to \{0, 1, \ldots, m_{i,k}\}$$

that associate with each mode j of each target t the return number $a_{mr_i}(t, j)$ from sensor i.

For each association, let $\Theta_{a_{mr_{ms}}}(k)$ and each $\Theta_{a_{mr_i}}(k)$ denote the events that $a_{mr_{ms}}$ and each a_{mr_i}, respectively, are the correct associations for the kth observation. For $1 \le t \le T$, $1 \le j \le n_t$, and $\mathcal{L} = (l_1, l_2, \ldots, l_{N_s})$ where $0 \le l_1 \le m_{1,k}, \ldots, 0 \le l_{N_s} \le m_{N_s,k}$, let $\theta^{t,j}_{\mathcal{L}}(k)$ denote the event that $a_{mr_{ms}}(t, j) = \mathcal{L}$ for the correct association $a_{mr_{ms}}$ for the kth observation. For $1 \le t \le T$, $1 \le j \le n_t$, $1 \le i \le N_s$, and $0 \le l \le m_{i,k}$, let $\theta^{t,j}_{l,i}(k)$ indicate the event that $a_{mr_i}(t, j) = l$ for the correct association a_{mr_i} of sensor i for the kth observation. The events $\theta^{t,j}_{\mathcal{L}}$ and $\theta^{t,j}_{l,i}$ are disjoint unions of events $\Theta_{a_{mr_{ms}}}$ and $\Theta_{a_{mr_i}}$, respectively. Let $\chi^{t,j}_{l,i}$ and $\chi^{t,j}_{\mathcal{L}}$ denote the conditional probabilities of the events $\theta^{t,j}_{l,i}$ and $\theta^{t,j}_{\mathcal{L}}$, respectively, given $\mathcal{Z}^k$. The probability $\chi^{t,j}_{l,i}$ is the single-sensor modal association probability that report l of sensor i is associated with mode j of target t, and the event probabilities we seek are the multisensor modal association probabilities $\chi^{t,j}_{\mathcal{L}}$.

At every time step, measurements are received and gated with the n_t prediction modes for each target t. Each set $\mathcal{L}$ of returns from the N_s sensors that gate with a mode j of a target t will yield a new filtered estimate mode for target t. Thus, the number of modes for each target t may expand to up to $n_t \prod_{i=1}^{N_s}(m_{i,k} + 1)$ modes depending upon how many measurements gate with each of the modes of target t. The state estimates $\hat{\boldsymbol{x}}^{t,j}_{\mathcal{L}}(k \mid k)$ and covariances $\boldsymbol{P}^{t,j}_{\mathcal{L}}(k \mid k)$ for the new modes are computed based upon the state prediction $\hat{\boldsymbol{x}}^{t,j}(k \mid k-1)$ and the association of mode j of target t with the set of $\mathcal{L}$ returns from the N_s sensors. With measurement errors from one sensor being independent of those from other sensors, the $\chi^{t,j}_{\mathcal{L}}$ probability weightings for the new modes are calculated by

$$\begin{aligned}
\chi^{t,j}_{\mathcal{L}} &= P\left(\theta^{t,j}_{\mathcal{L}} \mid \mathcal{Z}^k\right) \\
&= \sum_{a_{mr_{ms}}\colon a_{mr_{ms}}(t,j)=\mathcal{L}} P\left(\Theta_{a_{mr_{ms}}} \mid \mathcal{Z}^k\right) \\
&= \sum_{a_{mr_{ms}}\colon a_{mr_{ms}}(t,j)=\mathcal{L}} \prod_{i=1}^{N_s} P\left(\Theta_{a_{mr_i}} \mid \mathcal{Z}^k\right) \\
&= \sum_{a_{mr_1}\colon a_{mr_1}(t,j)=l_1, \ldots, a_{mr_{N_s}}\colon a_{mr_{N_s}}(t,j)=l_{N_s}} \prod_{i=1}^{N_s} P\left(\Theta_{a_{mr_i}} \mid \mathcal{Z}^k\right) \\
&= \prod_{i=1}^{N_s} \sum_{a_{mr_i}\colon a_{mr_i}(t,j)=l_i} P\left(\Theta_{a_{mr_i}} \mid \mathcal{Z}^k\right) \\
&= \prod_{i=1}^{N_s} \chi^{t,j}_{l_i,i}
\end{aligned} \tag{37}$$

Thus, the multisensor event probability $\chi^{t,j}_{\mathcal{L}}$ is the product of the single-sensor event probabilities $\chi^{t,j}_{l_i,i}$, which can be calculated using Eqs. (35) and (36). Before another set of measurements is received from the N_s sensors, the MR algorithm of Sec. III is applied to each target t to reduce the number of modes for each target to less than or equal to N_T.

Although the multisensor association probabilities $\beta^t_{\mathcal{L}}$ are not needed for computing the modal probabilities $\chi^{t,j}_{\mathcal{L}}$, it is interesting to note that the multisensor $\beta^t_{\mathcal{L}}$ have a similar property as with the multisensor $\chi^{t,j}_{\mathcal{L}}$:

$$\begin{aligned}
\beta^t_{\mathcal{L}} &= P\left(\theta^t_{\mathcal{L}} \mid \mathcal{Z}^k\right) \\
&= \sum_{a_{ms}\colon a_{ms}(t)=\mathcal{L}} P\left(\Theta_{a_{ms}} \mid \mathcal{Z}^k\right) \\
&= \sum_{a_{ms}\colon a_{ms}(t)=\mathcal{L}} \prod_{i=1}^{N_s} P\left(\Theta_{a_i} \mid \mathcal{Z}^k\right) \\
&= \prod_{i=1}^{N_s} \sum_{a_i\colon a_i(t)=l_i} P\left(\Theta_{a_i} \mid \mathcal{Z}^k\right) \\
&= \prod_{i=1}^{N_s} \beta^t_{l_i,i}
\end{aligned} \tag{38}$$

Analogous to the $\chi^{t,j}_{\mathcal{L}}$ and $\chi^{t,j}_{l_i,i}$ event probabilities, the multisensor event probability $\beta^{t,j}_{\mathcal{L}}$ is the product of the single-sensor MTMR event probabilities $\beta^{t,j}_{l_i,i}$.

VI. Simulation Evaluations

We have run simulations evaluating the MTMR and MSMTMR algorithms of Sec. IV and V. Simulations have been run for the tracking of two targets. The target dynamics are governed by Eq. (1) with time-invariant $\boldsymbol{F}$ and $\boldsymbol{G}$ matrices given by

$$\boldsymbol{F}^t = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{G}^t = \begin{bmatrix} \Delta^2/2 & 0 \\ \Delta & 0 \\ 0 & \Delta^2/2 \\ 0 & \Delta \end{bmatrix}$$

for $t = 1, 2$. The state vectors $\boldsymbol{x}^t(k) = [x \quad \dot{x} \quad y \quad \dot{y}]^T(k)$ represent the positions and velocities of the targets at time $k\Delta$, where Δ is the time step between measurements. The two targets are initially d units apart and initially move in parallel directions with the same speeds, but due to process or acceleration noise, their directions and speeds may change with time. The process noise vectors $\boldsymbol{w}^t(k)$ are independent 2×1 Gaussian noise vectors with constant covariance matrices

$$\boldsymbol{Q} = \begin{bmatrix} q & 0 \\ 0 & q \end{bmatrix}$$

For the simulations of the MTMR algorithm, there is one sensor whose measurements are governed by Eq. (2) with

$$\boldsymbol{H}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

For the MSMTMR simulations, a second identical sensor with $\boldsymbol{H}_2 = \boldsymbol{H}_1$ is used. The measurement noise vectors $\boldsymbol{v}^t_i(k)$ are independent 2×1 Gaussian noise vectors with constant covariance matrices

$$\boldsymbol{R} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}$$

In the simulations, we assume that the initial states of the targets are perfectly known and that each target is always well inside the surveillance region. Track initiation and deletion procedures are not incorporated in the simulations in order to isolate the performance of the MR data association algorithms. For more complete tracking algorithms, previously developed track initiation and deletion

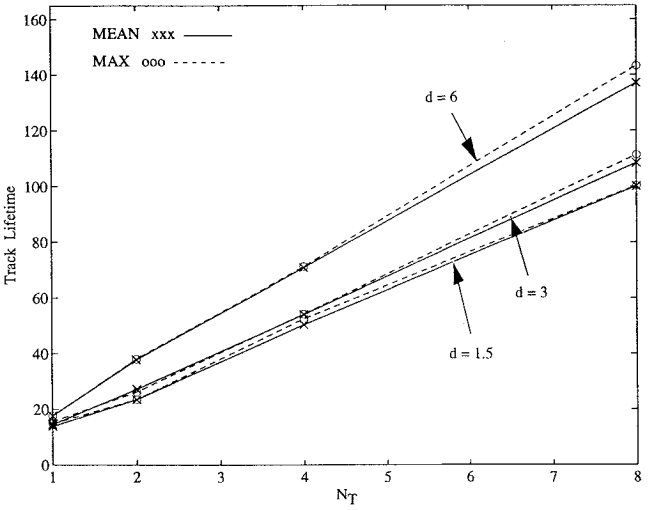

Fig. 1 Average track lifetime vs N_T for MTMR algorithm with two targets.

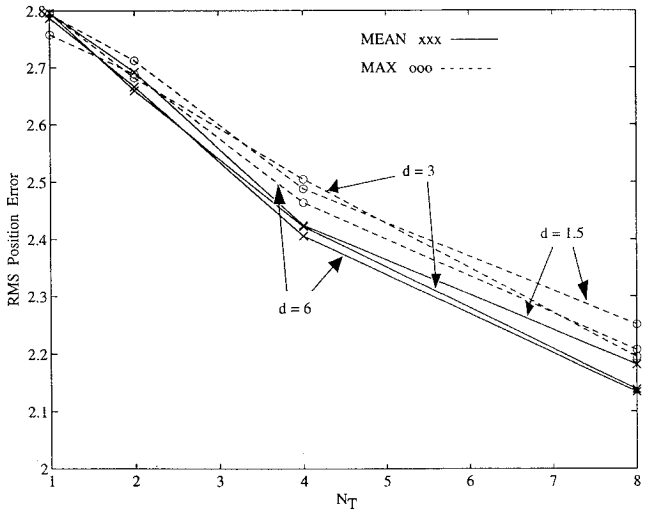

Fig. 2 Average rms error vs N_T for MTMR algorithm with two targets.

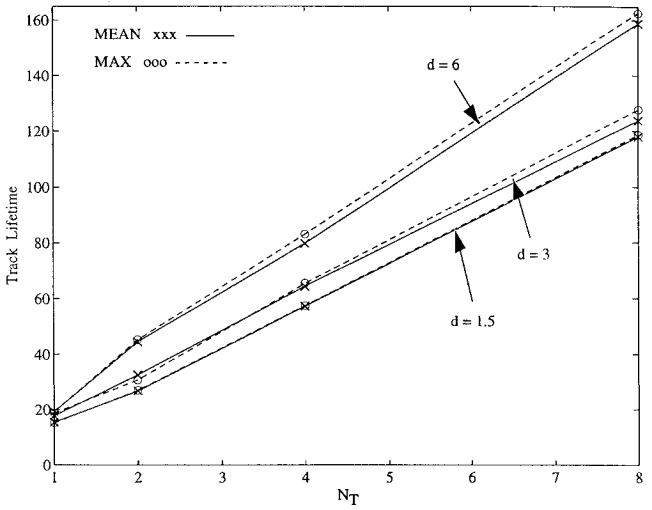

Fig. 3 Average track lifetime vs N_T for MSMTMR algorithm with two sensors and two targets.

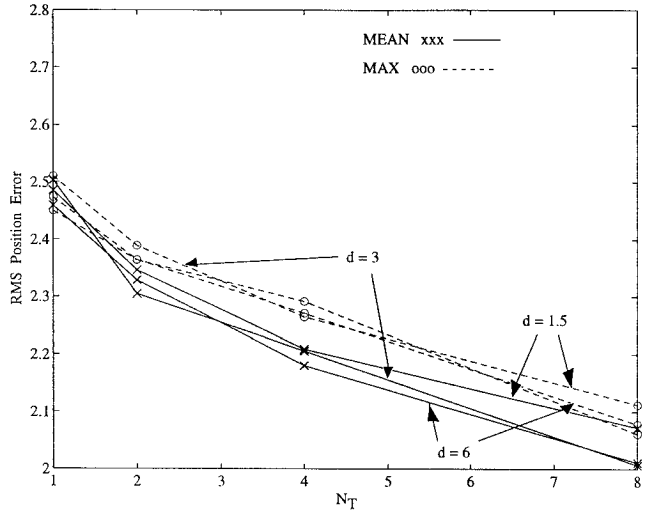

Fig. 4 Average rms error vs N_T for MSMTMR algorithm with two sensors and two targets.

techniques[1,2,9] can be adapted and combined with the MR algorithms. Separate track files would be kept and updated for tentative and confirmed tracks, and the MR data association and filtering algorithms would only be applied to the confirmed tracks.

For the simulation evaluations, false measurements are uniformly distributed throughout the surveillance region with density λ per unit area. Before data association, a gating procedure for each target is used to eliminate measurements that are not likely (with probability less than $1 - P_G$) to originate from the target. Given the gated (true and false) measurements, the MTMR and MSMTMR tracking filters compute probabilities of each of the measurements being a true target measurement. Estimates of the target state vectors are then calculated. Two methods for computing the estimates from the mixture-reduced modes are used. One computes a weighted average of the modes (which is the mean of the distribution) (MEAN), and the other takes the mode with the largest probability weighting (MAX). Finally, two measures of tracking performance are used. One is the average root-mean-square (rms) position error, and the other is a track lifetime metric. The track lifetime is defined[5] as the time until either i) the true measurement is rejected by the gating procedure for five consecutive time steps or ii) either the x or y coordinate of the filter estimate (using the MEAN or MAX method) is more than 10 standard deviations from the corresponding coordinate of the true target position for five consecutive time steps. Here, the standard deviations are those of the position estimates of the equivalent Kalman filters when there is no clutter.

In the simulations, the following parameter values were used: $\Delta = 1$, $q = 0.0625$, $r = 0.0625$, $\lambda = 0.2$, and $P_G = 0.999$. The parameters N_T and d were varied. With the above parameter values, the expected number of false measurements per gate, using steady-state Kalman filter covariances, is 2.1701 for the MTMR algorithm and 1.6639 for the MSMTMR algorithm. These numbers are different due to different steady-state covariances for the two algorithms. Although there are fewer expected clutter points per gate for the MSMTMR algorithm, the conditions of the tracklife metric are also stricter.

To evaluate the performances of the MTMR and MSMTMR algorithms, 20 sets of trajectories with associated measurements were generated and the algorithms were run until both tracks were lost. The track lifetimes were then averaged over both targets and across all 20 runs. Further, in each run, the rms position error of the track estimates from the truth were computed up to the time of track loss for each target. The rms position errors were then averaged across both targets over all 20 runs.

Simulation results for the MTMR algorithm are plotted in Figs. 1 and 2. In the plots, crosses and circles indicate actual simulation points. Figure 1 shows the average track lifetimes as N_T is varied for different initial separation distances d for both methods of estimation, MEAN and MAX. Figure 2 shows the corresponding average rms position errors for the various cases. Figures 3 and 4 show similar results for the MSMTMR algorithm when two sensors are used.

With $N_T = 1$, the MTMR algorithm reduces to the JPDA and the MSMTMR algorithm reduces to the MSJPDA. As N_T is increased (more modes to approximate each mixture of Gaussians), we see

that (as expected) the performances of the MTMR and MSMTMR algorithms improve. In Figs. 1 and 3, the track lifetimes increase with N_T as well as with the initial target separation d. For small initial target separations, reports due to one target may "interfere" with the estimation of the state of the other target. For initial target separations of $d \geq 6$, the tracking problem for the 20 sets of trajectories almost always reduces to tracking each target separately. Thus, for $d \geq 6$, the track lifetimes do not increase much further beyond the $d = 6$ curves.

From Figs. 2 and 4, we see that the average rms position errors generally decrease as N_T increases. Further, we see that, in general, the errors for $d = 6$ are smaller than those for $d = 3$, which are in turn smaller than for $d = 1.5$.

Comparing the results of the MEAN and MAX estimation schemes in the various figures, we see that the rms position errors for the MAX method are generally larger (about 1.8%) than for the MEAN method. However, the track lifetimes for the MAX scheme are slightly longer (about 2.0%).

Figures 1 and 3 and 2 and 4 are plotted on the same axis scales, respectively. Comparing Figs. 1 and 3, the average track lifetimes using the MSMTMR algorithm can be seen to be about 15% longer than those of the MTMR algorithm. Comparing Figs. 2 and 4, we see that the average rms error of the MTMR algorithm is about 11% larger than that of the MSMTMR algorithm for both the MEAN and MAX estimation schemes. Thus, due to the smaller rms errors and longer track lifetimes, use of an additional sensor in the MSMTMR algorithm yields better tracking results than using a single sensor in the MTMR algorithm.

Hence, we see the improved tracking performance for both the multitarget and multisensor extensions of the MR algorithm. As with the simulation results of the single-sensor mixture reduction algorithm,[5,7] we would expect the performance increases of the multitarget and multisensor extensions to taper off and approach the performances of the optimal multitarget and multisensor Bayesian filters as N_T is further increased beyond 8 (maximum N_T of simulations for Figs. 1–4). (Further simulations for larger N_T were not run due to project time constraints and the time required to run these computational complex algorithms for larger values of N_T.)

The multitarget and multisensor mixture reduction algorithms are extensions of the multitarget and multisensor probabilistic data association algorithms. Since several modes or hypotheses are allowed to represent each target, the MR algorithms are similar to track-oriented multiple hypothesis tracking (MHT) algorithms.[9] However, the modes are merged in the MR algorithms, whereas in multiple hypothesis tracking algorithms, the hypotheses or modes are pruned to reduce computational complexity. The merging technique of the MR approach reduces the possibility of discarding information (the true report) that originated from a target. Analytically comparing these two complex algorithms is difficult, however, and careful comparisons for particular applications need to be carried out to determine the more suitable algorithm to use. Another advantage of the MR approach is that there are fixed memory and computational requirements for a given number of targets and sensors and maximum number of modes, N_T, per target. In MHT algorithms, oftentimes the pruning is based on a probability threshold where modes or hypotheses with probabilities less than the threshold are pruned. In such algorithms, the maximum number of modes per target may vary from interval to interval, and hence memory and computational requirements will vary.

VII. Conclusions

A single-sensor, single-target MR algorithm has been extended both for multitarget tracking and for multisensor, multitarget tracking. The extension of the MR algorithm to the multitarget case is similar to that of extending the PDA algorithm for multitarget tracking in that the measurement-to-target association probabilities are computed jointly across all targets. Assuming that measurement errors are independent across sensors, extending the multitarget MR algorithm to the multisensor, multitarget MR algorithm results in the multisensor modal association probabilities being the product of the single-sensor modal association probabilities.

Just as the single-sensor, single-target MR algorithm is a generalization of the PDA filter, the multitarget, multisensor MR algorithms are generalizations of the multitarget, multisensor PDA algorithms, respectively. The MR approach may allow more modes to approximate each Gaussian mixture than the single mode of the PDA approach. Simulation results showed that as the maximum number of modes allowed to approximate each Gaussian mixture is increased, the performances of the multitarget, multisensor, MR algorithms improve beyond those of the corresponding multitarget, multisensor PDA algorithms. Extensive Monte Carlo simulations for different types of scenarios to more clearly and conclusively delineate the performances of the multitarget, multisensor MR algorithms and compare these algorithms with the PDA algorithm, nearest neighbor, and other approaches to tracking are a subject of further work.

The MR extensions have been formulated to yield association probability equations that have a similar structure as for the JPDA filter. Thus, the computation of the association probabilities needed for the new multitarget, multisensor MR algorithms can take advantage of fast JPDA algorithms that have been developed. However, the complexity of the multitarget, multisensor MR algorithms does grow exponentially with the number of targets and sensors, and more careful analysis of the computational complexity and additional ways to reduce the complexity of these algorithms are areas of future work. For the multisensor algorithm, implementing the multitarget MR filters sequentially on each sensor's data should yield some computational savings over implementing the multisensor filter in parallel on all the sensors' data as described in this paper.

Acknowledgments

Part of this work was performed under MITRE's Mission Oriented Investigation and Experimentation Program, Air Force contract F19628-89-C-0001. The author thanks Sean D. O'Neil of the MITRE Corporation, Bedford, MA, for many stimulating discussions on the mixture reduction method and on tracking in general. The author also thanks the anonymous reviewers for many constructive comments.

References

[1]Bar-Shalom, Y., and Fortmann, T. E., *Tracking and Data Association*, Academic, San Diego, 1988.

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

[3]Bogler, P. L., *Radar Principles with Applications to Tracking Systems*, Wiley, New York, 1990.

[4]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. 8, No. 3, 1983, pp. 173–183.

[5]Salmond, D. J., "Mixture Reduction Algorithms for Target Tracking in Clutter," *SPIE Signal and Data Processing of Small Targets*, Vol. 1305, 1990, pp. 434–445.

[6]O'Neil, S. D., and Pao, L. Y., "Multisensor Fusion Algorithms for Tracking," *Proceedings of the 1993 American Control Conference*, San Francisco, CA, 1993, pp. 859–863.

[7]Salmond, D. J., "Tracking in Uncertain Environments," Ph.D. Dissertation, Royal Aerospace Establishment, London, 1989.

[8]O'Neil, S. D., and Bridgland, M. F., "Fast Algorithms for Joint Probabilistic Data Association," *Proceedings of the National Symposium on Sensor Fusion*, Orlando, FL, 1991.

[9]Bar-Shalom, Y. (ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, Norwood, MA, 1990.