Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Distributed Multi-Sensor Multi-Target Tracking with Feedback†

Weerawat Khawsuk‡ and Lucy Y. Pao§

Abstract— **An algorithm that incorporates feedback in distributed fusion architectures for maintaining target tracks in cluttered environments is proposed. The decorrelated feedback sequences are constructed by compensating global updated estimates with track information due to global predicted estimates and local track estimates. Because of its orthogonal properties, these feedback sequences are then used in the filtering process to update local estimates before local processors acquire new sets of measurements. The process of constructing these feedback sequences is presented and implemented on a proposed distributed fusion system where each local processor receives measurements from multiple sensors.**

I. INTRODUCTION

More accurate and robust schemes to maintain trajectories of multiple targets in complex environments are increasingly desired for many applications such as air traffic control, military surveillance, and mobile robots. A centralized processing architecture is often assumed in mathematically developing tracking algorithms, and it has been shown that tracking performance of centralized fusion architectures improve significantly when multiple sensors are used [13]. However, increasing the number of sensors incurs a larger computational burden on the central processor as well as greater communication bandwidth requirements. In practice, distributed processing architectures are used due to their lower computational demands, lower communication bandwidth requirements, and greater reliability and survivability [5], [14], [15].

The general distributed fusion architecture of Figure 1 consists of several local processors and one global processor. Bi-directional communication between each level implies that the distributed fusion architecture possibly uses feedback in order to improve overall tracking performance. Each local processor independently tracks targets in its surveillance region with its own sensors. Measurements from different sensors are received simultaneously. Because of the uncertainty of measurement origin, many centralized data association algorithms such as Nearest Neighbor (NN) [2], Joint Probabilistic Data Association (JPDA) [2], or Mixture Reduction (MR) [13], [16] can be implemented on each local processor. The target state estimates from each local processor are then passed to a global processor and possibly other local processors. At the global processor, a distributed fusion algorithm employs track fusion to combine the local tracks to form global tracks of targets in the entire surveillance region.

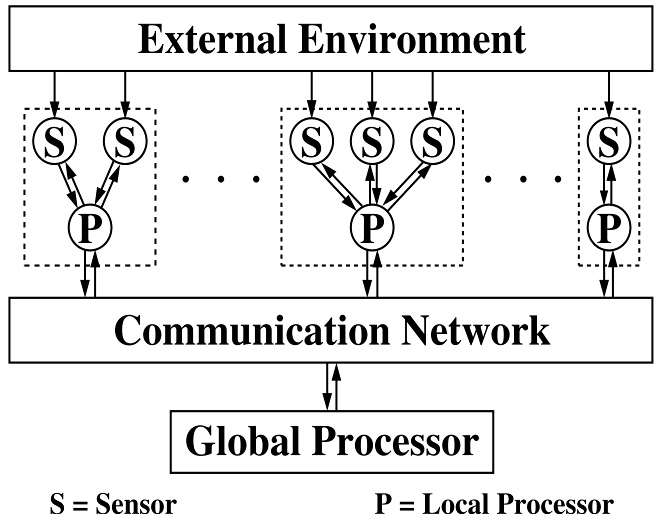

Figure 1. Distributed sensor fusion architecture.

One drawback of distributed fusion is the difficulty in merging state estimates from different local processors due to the loss of information inherent in forming the local track estimates. Several track fusion techniques such as track-to-track fusion [1], [3], [4] and decorrelation of state estimates [6], [9], [10], [11], [12] have been developed for various assumptions and configurations of distributed architectures. For track-to-track fusion, state estimates for a common target from different local processors are correlated, and computation of this correlation is rather cumbersome [1], [3]. Though an "optimal" track-to-track fusion is introduced [4], it does not incorporate any data association method.

Decorrelation techniques produce decorrelated sequences so that the global processor can process them as measurement inputs to a filtering algorithm [6], [9]. In this way, many well-known "centralized" processing algorithms can be utilized at the global level in a similar manner as in the local level. It can be shown that the "optimal" track-to-track fusion [4] and decorrelation techniques [6], [9] are mathematically equivalent when there is no measurement origin uncertainty. We have recently extended the decorrelated sequence approach [10], [11], [12] for more complex tracking environments that include the existence of clutter, data association, interacting targets, and multiple sensors where the sequential Multi-Sensor Joint Probabilistic Data Association (MSJPDA) algorithm is used at the local and global processors.

In general, the tracking performance of the global processor depends largely on the accuracy of the local track estimates and the track lifetimes achievable by each local processor. To improve the tracking performance at the local level and hence at the global level, in this paper, we incorporate

† This work was supported in part by the Office of Naval Research (Grant N00014-02-1-0136).

‡ Instructor in the Department of Electrical Engineering, Chulachomklao Royal Military Academy, Nakhon Nayok, Thailand, 26000. Recently graduated from the University of Colorado at Boulder, khawsuk@colorado.edu.
§ Associate professor in the Electrical and Computer Engineering Department, University of Colorado, Boulder, CO 80309, pao@colorado.edu.

a feedback process for the distributed fusion architecture. Previous work [19] on track-to-track fusion with a particular distributed fusion system using pure Kalman filters (where there is no measurement uncertainty) shows that feedback improves tracking performance of all local processors by reducing uncertainty in their updated state estimates. Thus, this paper extends the decorrelated sequence approach [10], [11] and derives how to construct decorrelated feedback sequences at the global level to pass back to each local processor for distributed multi-sensor multi-target tracking in cluttered environments where the MSJPDA algorithm is used. How these feedback sequences are used at each local processor is also addressed.

The sequential MSJPDA algorithm is reviewed in Section II, while the new decorrelated sequence algorithm incorporating feedback is presented in Section III. The process for constructing the feedback sequences is derived and its implementation at each local processor is discussed in Section IV. Finally, simulation results and concluding remarks are given in Sections V and VI.

II. Sequential MSJPDA Filter

The sequential MSJPDA filter is an algorithm for estimating stochastic systems such as in target tracking using multiple sensors. Let $x^t(k)$ denote the state vector of target t at the kth time interval. Suppose the target dynamics are determined by known $F^t(k)$ and $G^t(k)$ matrices and a random process noise vector $w^t(k)$ as follows:

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

where the noise vector $w^t(k)$ is a stochastically independent Gaussian random variable with zero mean and known covariance matrix, denoted as $\mathcal{N}[0, Q^t(k)]$. For a tracking system with N_s sensors, the tth target originated measurement $z_j^t(k)$ from the jth sensor is determined by a known matrix $H_j^t(k)$ and a random noise vector $v_j^t(k)$ as

$$z_j^t(k) = H_j^t(k)x^t(k) + v_j^t(k), \tag{2}$$

where the sensor noise vector $v_j^t(k)$ is also a stochastically independent Gaussian random variable with zero mean and known covariance matrix, denoted as $\mathcal{N}[0, R_j^t(k)]$.

The predicted state and its error covariance are

$$\hat{x}^t(k|k-1) = F^t(k)\hat{x}^t(k-1|k-1), \tag{3}$$

$$\begin{aligned} P^t(k|k-1) = {} & F^t(k)P^t(k-1|k-1)\left(F^t(k)\right)' \\ & + G^t(k)Q^t(k)\left(G^t(k)\right)'. \end{aligned} \tag{4}$$

In the sequential implementation, measurements from each sensor are processed one sensor at a time [18] and the algorithm can be summarized as follows. The predicted measurements, innovation sequence (measurement residual), and innovation covariance are

$$\hat{z}_j^t(k) = H_j^t(k)\hat{x}_{j-1}^t(k|k), \tag{5}$$

$$\nu_j^t(k) = z_j^t(k) - \hat{z}_j^t(k), \tag{6}$$

$$S_j^t(k) = H_j^t(k)P_{j-1}^t(k|k)\left(H_j^t(k)\right)' + R_j^t(k), \tag{7}$$

for $j = 1, 2, \ldots, N_s$. Note that $\hat{x}_0^t(k|k) = \hat{x}^t(k|k-1)$ and $P_0^t(k|k) = P^t(k|k-1)$ are the predicted state and its covariance, respectively. Then, the intermediate state estimates, Kalman gains, and state estimate covariances are computed as

$$\hat{x}_j^t(k|k) = \hat{x}_{j-1}^t(k|k) + K_j^t(k)\nu_j^t(k), \tag{8}$$

$$K_j^t(k) = P_{j-1}^t(k|k)\left(H_j^t(k)\right)'\left[S_j^t(k)\right]^{-1}, \tag{9}$$

$$P_j^t(k|k) = \left[I - K_j^t(k)H_j^t(k)\right]P_{j-1}^t(k|k). \tag{10}$$

The final state updates and covariances are obtained after processing measurements from the last sensor as

$$\hat{x}^t(k|k) = \hat{x}_{N_s}^t(k|k) \quad \text{and} \quad P^t(k|k) = P_{N_s}^t(k|k).$$

Once the state estimates and covariances are updated, the algorithm is repeated for the new set of measurements at the next time step.

When tracking targets in cluttered environments where the origin of measurements is not known, a data association algorithm such as the JPDA [2] method is needed. Clutter refers to detections or returns from background noise, false alarms, electromagnetic interference, neighboring targets, *etc*. A common mathematical model for such interference is a uniform distribution with density λ in the measurement space. These additional detections lead to the occurrence of several measurements in the validation region of each target. In the JPDA algorithm, the combined measurement

$$z_j^t(k) = \sum_{\ell=0}^{m_j(k)} \beta_{j,\ell}^t(k)z_{j,\ell}(k) \tag{11}$$

is used in (6) where $z_{j,\ell}(k)$ is the ℓth measurement for sensor j at time k, $\beta_{j,\ell}^t(k)$ is the probability that $z_{j,\ell}(k)$ is the measurement originating from target t, and $m_j(k)$ is the number of gated measurements from sensor j at time k. When $\ell = 0$, it denotes the possibility that there are no target-originated measurements and $z_{j,0}(k) = \hat{z}_j^t(k)$. The combined innovation is then used in the sequential MSJPDA filter [8] to update the state covariance as

$$\begin{aligned} P_j^t(k|k) = {} & \beta_{j,0}^t(k)P_{j-1}^t(k|k) \\ & + \left[1 - \beta_{j,0}^t(k)\right]\bar{P}_j^t(k|k) + \tilde{P}_j^t(k), \\ \bar{P}_j^t(k|k) = {} & \left[I - K_j^t(k)H_j^t(k)\right]P_{j-1}^t(k|k), \\ \tilde{P}_j^t(k) = {} & K_j^t(k)\left\{\sum_{\ell=0}^{m_j(k)} \beta_{j,\ell}^t(k)\nu_{j,\ell}^t(k)\left(\nu_{j,\ell}^t(k)\right)'\right. \\ & \left. - \nu_j^t(k)\left(\nu_j^t(k)\right)'\right\}\left(K_j^t(k)\right)', \end{aligned} \tag{12}$$

where $\nu_{j,\ell}^t(k) = z_{j,\ell}(k) - \hat{z}_j^t(k)$ is an individual innovation for target t due to measurement $z_{j,\ell}(k)$.

III. Decorrelated State Estimates

Decorrelation is a process of removing correlations between any 2 correlated input sequences to produce uncorrelated output sequences. Mathematically, it can be described

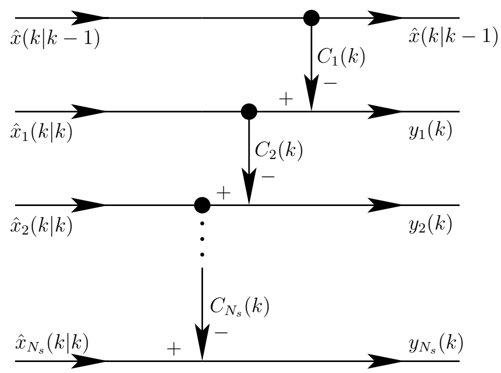

Figure 2. The channel model analysis for constructing decorrelated sequences for the sequential filtering of multiple sensors.

by the Gauss-Markov theorem if the statistics of these inputs are jointly normally distributed [17]. In our distributed tracking application, the correlated inputs are the predicted and updated state estimates of the filtering algorithm. For environments without clutter, the statistics of the predicted and updated states are indeed jointly normally distributed. However, the jointly normal distribution is no longer accurate for cluttered environments because the statistics of the track estimates and predictions are mixtures between jointly normal distributions from actual measurements and uniform distributions from clutter. Nevertheless, the decorrelation process can still be used to construct decorrelated output sequences if the cross correlation between any inputs is correctly identified [10], [12].

Using the sequential MSJPDA algorithm at each local processor, the decorrelation process at stage $j = 1, 2, \ldots, N_s$ of any local processor can be described as [11], [12]:

$$y_j(k) = \hat{x}_j(k|k) - C_j(k)\hat{x}_{j-1}(k|k), \quad (13)$$

$$C_j(k) = P_j(k|k)P_{j-1}^{-1}(k|k), \quad (14)$$

$$Y_j(k) = P_j(k|k) - P_j(k|k)P_{j-1}^{-1}(k|k)P_j(k|k), \quad (15)$$

where $\hat{x}_j(k|k)$ and $P_j(k|k)$ are, respectively, the intermediate state update and its corresponding error covariance matrix. When tracking in cluttered environments, $P_j(k|k)$ is computed using (12) instead of (10). $C_j(k)$ is a decorrelation matrix and $Y_j(k)$ is the covariance matrix of the decorrelated sequences $y_j(k)$. For simplicity, the superscripts t for target identity are dropped in the above equations. Figure 2 illustrates the decorrelation process for a local processor with N_s sensors.

The decorrelated sequences $\{y_1(k), \ldots, y_{N_s}(k)\}$ are orthogonal [11], [12], *i.e.*, they are all uncorrelated to each other, and remain uncorrelated for all time k. Having characteristics similar to those of actual measurements $z_j(k)$, the decorrelated sequences $y_j(k)$ are used as measurements for the global processor.

IV. Feedback

Track estimates at the global processor are at least as accurate as local estimates because they are obtained by combining state estimates or decorrelated sequences from all of the local processors. Tracking performance for the global processor will further improve if the local track estimates used in the combining process are more accurate. One practical approach to enhance global performance is to incorporate feedback from the global processor back to the local level. These feedback sequences are then used to update the local estimates before each local processor acquires new sets of measurements from their own sensors. Previous analysis [7], [19] indicates that a specific distributed tracking system using track-to-track fusion with feedback indeed improves local tracking performance because it reduces the error covariance of each local estimates. These prior developments, however, are based only on the pure Kalman filter without incorporating any data association method.

One plausible choice of feedback information is the global intermediate estimates $\hat{x}_{G,j}(k|k)$ obtained from the sequential MSJPDA algorithm. These estimates, however, are correlated with each local state estimate $\hat{x}_p(k|k)$. To bypass this correlation problem, the decorrelation technique is used to transform these global intermediate estimates into uncorrelated sequences. Passing all global decorrelated sequences as feedback to each local processor is rather a redundant and cumbersome process. It is obvious that the global decorrelated sequences already contain some knowledge of the local estimates. As the numbers of local processors and sensors increase, so does the number of decorrelated sequences. Thus, the desired feedback to a particular local processor should contain only current track information from all other local processors [6], [7].

The overall feedback process consists of 3 stages: (a) decorrelation of global state estimates, (b) construction of feedback sequences, and (c) incorporation of these sequences at the local processors. Once intermediate estimates of the global processor $\hat{x}_{G,j}(k|k)$ are obtained, a similar process as described in (13) – (15) is used to construct the global decorrelated sequences $y_{G,j}(k)$, their decorrelation matrices $C_{G,j}(k)$, and their corresponding measurement matrices $B_{G,j}(k) = I - C_{G,j}(k)$. The feedback sequences for a particular local processor are formulated and then passed to that local processor for further processing.

A. Derivations and Constructions

The superscript t for target identity is omitted for simplicity. The feedback sequences $y_{f,p}(k)$ for local processor p can be formulated by removing the correlation due to predicted estimates $\hat{x}_G(k|k-1)$ and that due to global decorrelated sequences $y_{G,\ell}(k)$ of the same local processor from the current global state estimates $\hat{x}_{G,N}(k|k)$:

$$y_{f,p}(k) = \hat{x}_{G,N}(k|k) - C_{f,0}(k)\hat{x}_G(k|k-1) - \sum_{\ell>0} C_{f,\ell}(k)y_{G,\ell}(k). \quad (16)$$

Coefficients $C_{f,0}(k)$ and $C_{f,\ell}(k)$, respectively, are associated with $\hat{x}_G(k|k-1)$ and $y_{G,\ell}(k)$. The summation is over those indices ℓ for which $y_{G,\ell}(k)$ are decorrelated sequences resulting from track estimates of local processor p.

Since we can express the feedback sequence as $y_{f,p}(k) = B_{f,p}(k)x(k) - \tilde{y}_{f,p}(k)$, the feedback error is

$$\tilde{y}_{f,p}(k) = B_{f,p}(k)x(k) - \hat{x}_{G,N}(k|k) + C_{f,0}(k)\hat{x}_G(k|k-1) + \sum_{\ell>0} C_{f,\ell}(k) y_{G,\ell}(k). \quad (17)$$

Writing each estimated quantity in terms of its true state and its random error, $\tilde{y}_{f,p}(k)$ is expanded to

$$\tilde{y}_{f,p}(k) = \left[B_{f,p}(k) + C_{f,0}(k) + \sum_{\ell>0} C_{f,\ell}(k) B_{G,\ell}(k) - I\right] \cdot x(k) + \tilde{x}_{G,N}(k|k) - C_0(k)\tilde{x}_G(k|k-1) - \sum_{\ell>0} C_{f,\ell}(k)\tilde{y}_{G,\ell}(k). \quad (18)$$

To determine the unknown values of $C_{f,0}(k)$ and $C_{f,\ell}(k)$, the following assumptions are enforced:

(A1) All errors at time k have zero mean, *i.e.*, $E\{\tilde{y}_{f,p}(k)\}$, $E\{\tilde{x}_{G,N}(k|k)\}$, $E\{\tilde{x}_G(k|k-1)\}$, and $E\{\tilde{y}_{G,\ell}(k)\}$ are zero.

(A2) At any time k, there is no correlation among $\tilde{y}_{f,p}(k)$, $\tilde{y}_{G,\ell}(k)$, and $\tilde{x}_G(k|k-1)$, *i.e.*, $E\left\{\tilde{y}_{f,p}(k)\tilde{y}'_{G,\ell}(k)\right\}$, $E\left\{\tilde{y}_{f,p}(k)\tilde{x}'_G(k|k-1)\right\}$, $E\left\{\tilde{y}_{G,\ell}(k)\tilde{x}'_G(k|k-1)\right\}$, and $E\left\{\tilde{y}_{G,\ell}(k)\tilde{y}'_{G,i}(k)\right\}$ for $\ell \neq i$ are zero.

If all error sequences have zero mean, then the coefficient of $x(k)$ in (18) vanishes. Consequently, we obtain

$$B_{f,p}(k) = I - C_{f,0}(k) - \sum_{\ell>0} C_{f,\ell}(k) B_{G,\ell}(k), \quad (19)$$

and the feedback error in (18) becomes

$$\tilde{y}_{f,p}(k) = \tilde{x}_{G,N}(k|k) - C_{f,0}(k)\tilde{x}_G(k|k-1) - \sum_{\ell>0} C_{f,\ell}(k)\tilde{y}_{G,\ell}(k). \quad (20)$$

Since the construction of the feedback sequence $y_{f,p}(k)$ ensures that there is no correlation between $\tilde{x}_G(k|k-1)$ and $\tilde{y}_{f,p}(k)$, *i.e.*, $E\left\{\tilde{y}_{f,p}(k)\tilde{x}'_G(k|k-1)\right\} = 0$, we must have

$$0 = M_{N,0}(k) - C_{f,0}(k)P_G(k|k-1) - \sum_{\ell>0} C_{f,\ell}(k) E\left\{\tilde{y}_{G,\ell}(k)\tilde{x}'_G(k|k-1)\right\},$$
$$M_{N,0}(k) = C_{f,0}(k)P_G(k|k-1), \quad (21)$$

where $M_{N,0}(k) = E\left\{\tilde{x}_{G,N}(k|k)\tilde{x}'_G(k|k-1)\right\}$ and $P_G(k|k-1) = E\left\{\tilde{x}_G(k|k-1)\tilde{x}'_G(k|k-1)\right\}$. The last expectation vanishes due to the **(A2)** assumption. We know from [11], [12] that the global decorrelated error $\tilde{y}_{G,j}(k)$ can be expressed as a linear combination of $\tilde{x}_{G,j}(k|k)$, $\tilde{x}_G(k|k-1)$, and previous $\tilde{y}_{G,i}(k)$ for $i = 1, \ldots, j-1$ as

$$\tilde{y}_{G,j}(k) = \tilde{x}_{G,j}(k|k) - \sum_{i=1}^{j-1} D_{j,i}(k)\tilde{y}_{G,j-i}(k) - D_{j,j}(k)\tilde{x}_G(k|k-1), \quad (22)$$

where coefficients $D_{m,n}(k) \triangleq \prod_{r=1}^{n} C_{G,m-r+1}(k)$. Using the **(A2)** condition, *i.e.*, $E\left\{\tilde{y}_{G,N}(k)\tilde{x}'_G(k|k-1)\right\} = 0$, we obtain an alternative form of $M_{N,0}(k)$ as

$$0 = M_{N,0}(k) - D_{N,N}(k)P_G(k|k-1) - \sum_{i=1}^{N-1} D_{N,i}(k) E\left\{\tilde{y}_{G,N-i}(k)\tilde{x}'_G(k|k-1)\right\},$$
$$M_{N,0}(k) = D_{N,N}(k)P_G(k|k-1). \quad (23)$$

Comparing (21) with (23) and then using coefficients $C_{G,i}(k) = P_{G,i}(k|k)P^{-1}_{G,i-1}(k|k)$ resulting from the global decorrelation stage, we obtain

$$C_{f,0}(k) = D_{N,N}(k) = P_{G,N}(k|k)P_G^{-1}(k|k-1). \quad (24)$$

To determine $C_{f,\ell}(k)$ when $\ell \neq 0$, we enforce that there is no correlation among $\tilde{y}_{f,p}(k)$, $\tilde{y}_{G,\ell}(k)$, and $\tilde{x}_G(k|k-1)$. Thus, we obtain

$$0 = E\left\{\tilde{y}_{f,p}(k)\tilde{y}'_{G,\ell}(k)\right\} = M_{N,\ell}(k) - C_{f,0}(k)E\left\{\tilde{x}_G(k|k-1)\tilde{y}'_{G,\ell}(k)\right\} - \sum_{i>0} C_{f,i}(k) E\left\{\tilde{y}_{G,i}(k)\tilde{y}'_{G,\ell}(k)\right\}.$$
$$M_{N,\ell}(k) = C_{f,\ell}(k)Y_{G,\ell}(k), \quad (25)$$

where $M_{N,\ell}(k) = E\left\{\tilde{x}_{G,N}(k|k)\tilde{y}'_{G,\ell}(k)\right\}$ and $Y_{G,\ell}(k) = E\left\{\tilde{y}_{G,\ell}(k)\tilde{y}'_{G,\ell}(k)\right\}$. Similarly using (22) and the orthogonal properties, we obtain another expression for $M_{N,\ell}(k)$ as

$$0 = E\left\{\tilde{y}_{G,N}(k)\tilde{y}'_{G,\ell}(k)\right\} = M_{N,\ell}(k) - \sum_{i=1}^{N-1} D_{N,i}(k) E\left\{\tilde{y}_{G,N-i}(k)\tilde{y}'_{G,\ell}(k)\right\} - D_{N,N}(k)E\left\{\tilde{x}_G(k|k-1)\tilde{y}'_{G,\ell}(k)\right\},$$
$$M_{N,\ell}(k) = D_{N,N-\ell}(k)Y_{G,\ell}(k). \quad (26)$$

Comparing (25) with (26) and then using coefficients $C_{G,i}(k) = P_{G,i}(k|k)P^{-1}_{G,i-1}(k|k)$ from the global decorrelation stage, we obtain

$$C_{f,\ell}(k) = D_{N,N-\ell}(k) = P_{G,N}(k|k)P^{-1}_{G,\ell}(k|k). \quad (27)$$

Next, the error covariances of the feedback sequences for local processor p, denoted as $Y_{f,p}(k) = E\left\{\tilde{y}_{f,p}(k)\tilde{y}'_{f,p}(k)\right\}$ can be computed as follows

$$Y_{f,p}(k) = P_{G,N}(k|k) - M_{N,0}(k)C'_{f,0}(k) - \sum_{\ell>0} M_{N,\ell}(k)C'_{f,\ell}(k) - C_{f,0}(k)M'_{N,0}(k) + C_{f,0}(k)P_G(k|k-1)C'_{f,0}(k) - \sum_{\ell>0} C_{f,\ell}(k)M'_{N,\ell}(k) + \sum_{\ell>0} C_{f,\ell}(k)Y_{G,\ell}(k)C'_{f,\ell}(k), \quad (28)$$

where $P_{G,N}(k|k) = E\left\{\tilde{x}_{G,N}(k|k)\tilde{x}'_{G,N}(k|k)\right\}$.

We know from [11], [12] that $Y_{G,j}(k)$ obtained from the global decorrelation stage could be alternatively expressed as $Y_{G,j}(k) = B_{G,j}(k)P^t_{G,j}(k|k)$. Thus using (21), (24), (25), (27), and $Y_{G,\ell}(k)$, we have the following:

$$\begin{aligned} M_{N,0}(k)C'_{f,0}(k) &= C_{f,0}(k)P_G(k|k-1)C'_{f,0}(k) \\ &= C_{f,0}(k)P_{G,N}(k|k), \end{aligned} \quad (29)$$

$$\begin{aligned} M_{N,\ell}(k)C'_{f,\ell}(k) &= C_{f,\ell}(k)Y_{G,\ell}(k)C'_{f,\ell}(k) \\ &= C_{f,\ell}(k)B_{G,\ell}(k)P_{G,N}(k|k). \end{aligned} \quad (30)$$

Substituting (29) and (30) in (28) and after some cancellations, we obtain

$$\begin{aligned} Y_{f,p}(k) &= \left[I - C_{f,0}(k) - \sum_{\ell>0} C_{f,\ell}(k)B_{G,\ell}(k)\right] \\ &\cdot P_{G,N}(k|k) = B_{f,p}(k)P_{G,N}(k|k). \end{aligned} \quad (31)$$

B. Incorporation at the Local Processor

The global processor can construct the desired feedback sequences for local processor p with the following process:

$$\begin{aligned} y_{f,p}(k) &= \hat{x}_{G,N}(k|k) - C_{f,0}(k)\hat{x}_G(k|k-1) \\ &\quad - \sum_{\ell>0} C_{f,\ell}(k)y_{G,\ell}(k), \end{aligned} \quad (32)$$

$$C_{f,\ell}(k) = \begin{cases} P_{G,N}(k|k)P_G^{-1}(k|k-1), & \ell = 0, \\ P_{G,N}(k|k)P_{G,\ell}^{-1}(k|k), & \ell = 1,\ldots,N, \end{cases} \quad (33)$$

$$B_{f,p}(k) = I - C_{f,0}(k) - \sum_{\ell>0} C_{f,\ell}(k)B_{G,\ell}(k), \quad (34)$$

$$Y_{f,p}(k) = B_{f,p}(k)P_{G,N}(k|k). \quad (35)$$

The summation is over those indices ℓ for which $y_{G,\ell}(k)$ are decorrelated sequences resulting from track estimates of local processor p.

Once these feedback sequences are sent to their respective local processor p, they are used in the local data association and filtering algorithm to update local state estimates $\hat{x}_p(k|k)$. The feedback sequence $y_{f,p}(k)$ and its covariance $Y_{f,p}(k)$ are treated as a measurement and its equivalent noise covariance, respectively. The $B_{f,p}(k)$ matrix now serves as the measurement matrix for local processor p. In other words, the JPDA algorithm at each local processor uses the following substitutions

$$z(k) \Leftarrow y_{f,p}(k), \quad R(k) \Leftarrow Y_{f,p}(k), \quad H(k) \Leftarrow B_{f,p}(k).$$

This data association and filtering process must be completed before local processors acquire new sets of measurements from their own sensors, and the updated estimates resulting from the feedback sequences are used to initialize the filtering process of the next time step.

V. Monte-Carlo Simulations and Results

We have evaluated the incorporation of feedback using the decorrelated sequence method on distributed fusion architectures as in Figure 1. Monte-Carlo simulations of this fusion algorithm have been run for a wide range of configurations [12]. Here, we provide only a sampling of results for various two-sensor-per-local-processor configurations such as the (2,2), (2,2,2), and (2,2,2,2) configurations. For each configuration, the numbers listed in a sequence indicate the numbers of sensors for each local processor while a sum of the sequence represents the overall number of sensors for the distributed tracking system. The results for other configurations show similar trends [12].

A. Tracking Models

In the simulations, the distributed system tracks two independent targets moving in two dimensions in nominally straight lines corrupted by acceleration noise. The state vector of target t at the kth time interval consists of the position and velocity of the target in the x and y directions, *i.e.*, $x^t(k) = [\mathrm{x}\ \dot{\mathrm{x}}\ \mathrm{y}\ \dot{\mathrm{y}}]'$ for $t = 1, 2$. The system parameters for each target are identical and time-invariant with

$$F = \begin{bmatrix} 1 & \delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = q \cdot \begin{bmatrix} \delta^3/3 & \delta^2/2 & 0 & 0 \\ \delta^2/2 & \delta & 0 & 0 \\ 0 & 0 & \delta^3/3 & \delta^2/2 \\ 0 & 0 & \delta^2/2 & \delta \end{bmatrix}.$$

$$\begin{aligned} &H = I_{4\times 4}, \quad && Q = q \cdot I_{4\times 4}, \quad && R = q \cdot I_{4\times 4}, \\ &\delta = 1, && q = 0.0144. \end{aligned}$$

An identity measurement matrix implies that measurements of all target states are available at each local processor. The initial states are assumed to have Gaussian distributions with known means $\hat{x}^t(0|0)$ and covariances $P^t(0|0)$, with the initial positions of the two targets being 5 units apart in the y direction:

$$\hat{x}^1(0|0) = [5, 0.5, 5, 0.5]' \text{ and } \hat{x}^2(0|0) = [5, 0.5, 10, 0.5]'.$$

B. Performance Metrics

The performance measures used are the average RMS error of the estimated target states and the average track lifetime. Denote the error $e(k) = x(k) - \hat{x}(k|k)$ as the difference between the true state and the updated estimate obtained by the tracking algorithm, then the RMS error is computed by

$$RMS = \sqrt{\frac{1}{L}\sum(e(k))'e(k)},$$

where L is the number of "good" tracking points. These points are taken into account only when the tracking algorithm simultaneously satisfies the following criteria for a given target and a given processor:

(i). At least one true target-originated measurement from any of the sensors of the processor lies inside the gated region for that target.

(ii). The Mahalanobis distance of the target state error $d(k) = (e(k))'(P_a(k))^{-1}e(k) \leq 18.4668$, where $P_a(k)$ is the updated state covariance resulting from the Kalman filtering process.

The track lifetime of a target represents how long the tracking algorithm can maintain target trajectories. The time

TABLE I

AVERAGE NUMBER OF CLUTTER MEASUREMENTS PER GATE.

Clutter Density (λ)	Configurations			
	2-Sensor Centralized	(2,2)	(2,2,2)	(2,2,2,2)
0.25	1.434	1.355	1.316	1.296
0.35	2.287	2.033	1.963	1.927
0.45	3.429	2.904	2.732	2.653
0.55	4.783	3.793	3.581	3.477
0.65	6.033	4.945	4.498	4.372

at which the tracking algorithm gives the last accurate estimate is designated as the target track lifetime. The local processor is said to lose track of a target if one of the following criteria is true:

(iii). The true target-originated measurements from all sensors lie outside the gated region for 5 consecutive times.

(iv). The distance $d(k) > 18.4668$ for 5 consecutive times.

For the global processor, only the 2nd and 4th criteria above are used in the RMS error computation and the track lifetime determination, respectively. It is not possible to use either the 1st or 3rd criteria because the global processor does not have access to the measurements received by all sensors. Both RMS errors and track lifetimes are averaged over the number of observed targets for each simulation. For the local processor performance, these two metrics are further averaged over the number of local processors. The final average values are then again averaged over all 100 Monte Carlo simulations.

C. Simulation Results

We are interested in tracking targets in a cluttered environment. The number of clutter measurements is determined by the density λ which is varied between $\lambda = 0.25$ to $\lambda = 0.65$ for the two-sensor-per-local-processor distributed fusion architecture configurations. The average numbers of clutter measurements per gate are given in Table I for the central processor of a 2-sensor centralized configuration without feedback as well as the local processor of a two-sensor-per-local-processor distributed fusion architecture configuration with feedback. Implementation of feedback clearly yields smaller average numbers of clutter measurements per gate. As the number of local processors increases, the average number of clutter measurements per gate decreases for the same clutter density. The results imply that the size of the gated region for each sensor of the local processor is reduced when feedback is implemented. In other words, feedback reduces the error covariance of the local target estimates [19]. As a result, we should see improvement in the tracking performance.

Figure 3 shows the average RMS performance for both local and global processors when feedback is incorporated. The global average RMS error decreases dramatically as the number of local processors increases, while there is only

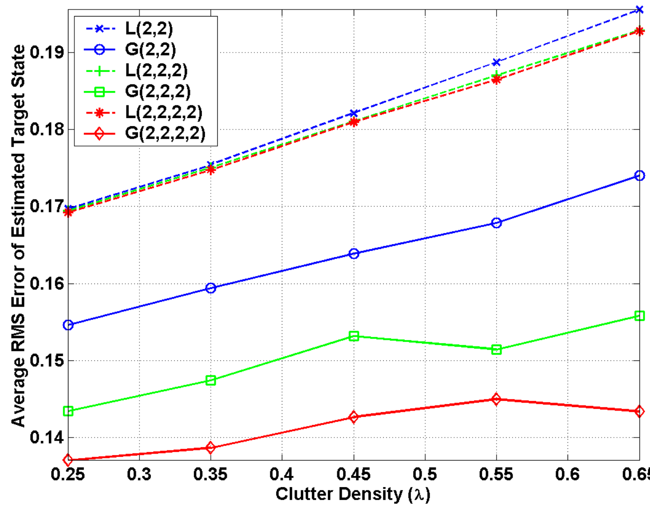

Figure 3. Average RMS error of the estimated target state versus clutter density λ for various two-sensor-per-local-processor distributed fusion architecture configurations with feedback implementation. L(2,2) and G(2,2) indicate the average RMS error at the local and global processors of a (2,2) feedback configuration, L(2,2,2) and G(2,2,2) indicate the average RMS error at the local and global processors of a (2,2,2) feedback configuration, L(2,2,2,2) and G(2,2,2,2) indicate the average RMS error at the local and global processors of a (2,2,2,2) feedback configuration.

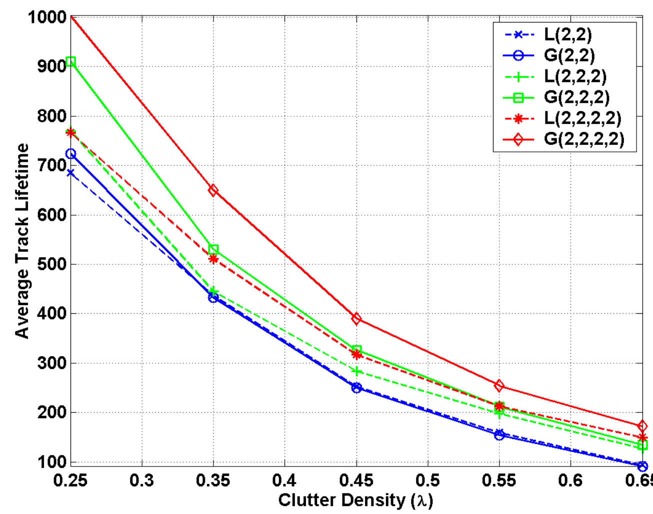

Figure 4. Average track lifetime versus clutter density λ for various two-sensor-per-local-processor distributed fusion architecture configurations with feedback implementation. L(2,2) and G(2,2) indicate the average track lifetime for the local and global processors of a (2,2) feedback configuration, L(2,2,2) and G(2,2,2) indicate the average track lifetime for the local and global processors of a (2,2,2) feedback configuration, L(2,2,2,2) and G(2,2,2,2) indicate the average track lifetime for the local and global processors of a (2,2,2,2) feedback configuration.

a minimal improvement for the local average RMS error. For the track lifetime performance shown in Figure 4, the different between the average track lifetime of the local and global processors becomes larger as the number of local processors increases.

Figures 5 and 6 demonstrate the performance comparison of distributed fusion architecture configurations with and without feedback. We see that incorporation of feedback improves both RMS error and track lifetime performance of

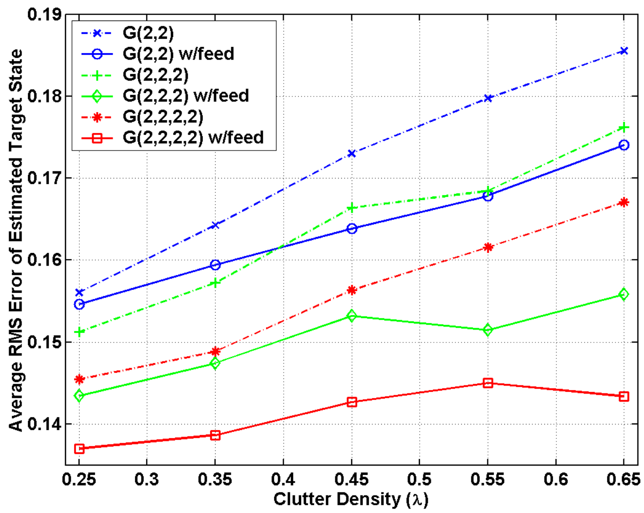

Figure 5. Average RMS error of the estimated target state versus clutter density λ for various two-sensor-per-local-processor distributed fusion architecture configurations with and without feedback.

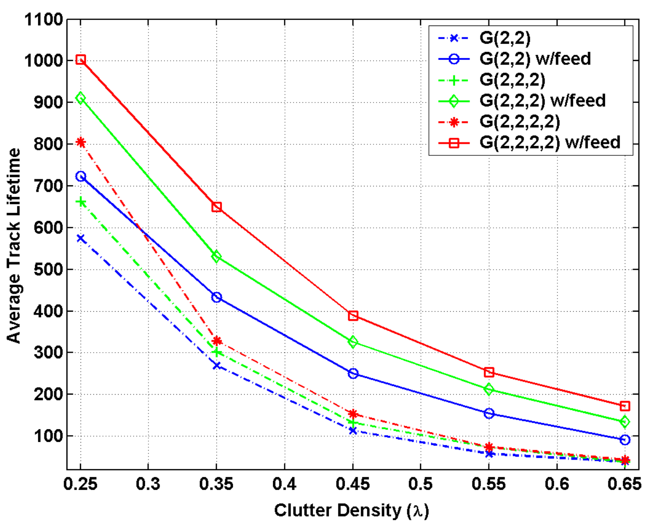

Figure 6. Average track lifetime versus clutter density λ for various two-sensor-per-local-processor distributed fusion architecture configurations with and without feedback.

the global processor. Adding more local processors yields a larger improvement in both average RMS error and track lifetime performance at the global processor when feedback is used rather than not used. As the clutter density increases, using feedback yields much lower average RMS error of the global processor than without using feedback. Furthermore, the simulation results from [12] show that the tracking performance of the local processors also improves noticeably when feedback is implemented.

VI. Conclusions

An algorithm to incorporate feedback for a general distributed fusion architecture has been developed. The construction of feedback sequences removes correlations due to the global predicted estimates and the decorrelated sequences of the local processors from the global state updates. This uncorrelated property allows the local processor to treat the feedback sequences as further measurements for processing, and the feedback sequences are processed before a new set of actual sensor measurements arrives. The decorrelated estimation technique for distributed fusion architectures shows significant improvement in tracking performance in terms of lower average RMS errors and longer average track lifetimes at both local and global processors when feedback is incorporated.

References

[1] Y. Bar-Shalom. "On the Track-to-Track Correlation Problem," *IEEE Trans. Automatic Control*, 26(2): 571–572, 1981.

[2] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*, Academic Press, Inc., San Diego, CA, 1988.

[3] C. B. Chang and L. C. Youens. "Measurement Correlation for Multiple Sensor Tracking in a Dense Target Environment," *IEEE Trans. Automatic Control*, 27(6): 1250–1252, 1982.

[4] K. C. Chang, R. K. Saha, and Y. Bar-Shalom. "On Optimal Track-to-Track Fusion," *IEEE Trans. Aerospace and Electronic Systems*, 33(4): 1271–1276, 1997.

[5] C. Y. Chong, K. C. Chang, and S. Mori. "Distributed Tracking in Distributed Sensor Networks," *Proc. American Control Conf.*, pp. 1863–1868, 1986.

[6] O. E. Drummond. "Feedback in Track Fusion Without Process Noise," *SPIE Signal and Data Processing of Small Targets*, 2561: 369–383, 1995.

[7] O. E. Drummond. "Track Fusion With Feedback," *SPIE Signal and Data Processing of Small Targets*, 2759: 342–360, 1996.

[8] C. W. Frei and L. Y. Pao. "Alternatives to Monte Carlo Simulation Evaluations of Two Multisensor Fusion Algorithms," *Automatica*, 34(1): 103–110, 1998.

[9] G. Frenkel. "Multisensor Tracking of Ballistic Targets," *SPIE Signal and Data Processing of Small Targets*, 2561: 337–346, 1995.

[10] W. Khawsuk and L. Y. Pao. "Decorrelated State Estimation for Distributed Tracking of Interacting Targets in Cluttered Environments," *Proc American Control Conf.*, pp. 899–904, 2002.

[11] W. Khawsuk and L. Y. Pao. "Decorrelated State Estimation for Distributed Tracking using Multiple Sensors in Cluttered Environments," *Proc. American Control Conf.*, pp. 3208–3214, 2003.

[12] W. Khawsuk. *Decorrelated State Estimation for Distributed Tracking in Cluttered Environments*, Ph.D. thesis, University of Colorado at Boulder, Feb. 2004.

[13] L. Y. Pao. "Centralized Multisensor Fusion Algorithms for Tracking Applications," *Control Engineering Practice*, 2(5): 875–887, 1994.

[14] L. Y. Pao. "Measurement Reconstruction Approach for Distributed Multisensor Fusion," *J. Guidance, Control, and Dynamics*, 19(4): 842–847, 1996.

[15] L. Y. Pao and M. K. Kalandros. "Algorithms for a Class of Distributed Architecture Tracking," *Proc. American Control Conf.*, pp. 1434–1438, 1997.

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

[17] L. L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*, Addison-Wesley, Reading, MA, 1991.

[18] D. Willner, C. B. Chang and K. P. Dunn. "Kalman Filter Algorithms for a Multi-Sensor System," *Proc. IEEE Conf. Decision and Control*, pp. 570–574, 1976.

[19] Y. Zhu, J. Zhao, K. Zhang, X. R. Li, and Z. You. "Performance Analysis for Distributed Track Fusion with Feedback," *Proc. World Congress on Intelligent Control and Automation*, pp. 2933–2936, 2000.