A Track-to-Track Fusion Method for Tracks with Unknown Correlations

Mahboubeh Zarei-Jalalabadi, Seyed M. B. Malaek, and Solmaz S. Kia, *Member, IEEE*

***Abstract*—This paper deals with the problem of track-to-track fusion under unknown correlations. We propose a novel method to construct the correlation terms between tracks from two sensors. We start by showing that the cross-covariance matrix of any two tracks can be expressed as the product of square roots of the tracks' covariance matrices and a contraction matrix. Then, we propose an optimization problem that obtains an estimate of this contraction matrix in a way that the fused track is less conservative than the one obtained by the well-known Covariance Intersection (CI) method but, at the same time, it is conservative in comparison with the optimal track obtained using the exact cross-covariance between the tracks. Through rigorous analysis we demonstrate our new fusion algorithm's properties. We also cast our design optimization problem as a difference of convex (DC) programming problem, which can be solved in an efficient manner using DC programming software solutions. We demonstrate our results through Monte Carlo simulations.**

***Index Terms*—Covariance Intersection, Difference of Convex Programming, Estimation, Sensor Fusion, Sensor Networks**

I. Introduction

NOWADAYS wireless sensors with embedded computing and communication capabilities play a vital role in provisioning real-time monitoring and control in many applications such as environmental monitoring, fire detection, object tracking, vehicular Ad-Hoc networks, and body area networks (see, e.g., [1] and references therein). The effectiveness and the safe operation of these applications rely on the accuracy of state estimates obtained from fusing information of the sensor stations. However, because of limitations such as network congestion and also energy constraint of wireless sensors, which are mostly battery-operated, the transmission of data from sensor stations to fusion centers is not usually possible in a persistent manner. Distributed solutions in which sensor stations maintain a local filter to process their local measurements and then combine their local tracks with the neighboring sensor stations are proposed as energy efficient solutions for wireless sensor networks. Proper track-to-track fusion is known to give more accurate estimates than using the information from a single sensor [2]–[4].

Due to the common process model, the local estimates of sensor stations are correlated [5], [6]. In earlier work on track-to-track fusion, this correlation was ignored [7]. However, it turned out that disregarding correlations results in fused estimates that tend to be too optimistic. This property is referred to as the inconsistency phenomenon, which has been shown to lead even to filter divergence [8], [9]. To overcome the problem of inconsistency, the correlations between the tracks should be taken into account in either explicit or implicit manner. Explicit account increases the complexity of distributed algorithms and require either multi-hop/all-to-all in-network communication [10] or high frequency communication among neighboring agents as in distributed Kalman-consensus algorithms (e.g. see [11], [12]). Consequently, there has been a great interest in track-to-track fusion methods that account for correlation terms in an implicit manner by conservatively bounding missing or discarded cross-covariance information. The prime example of such techniques is the *Covariance Intersection* (CI) method [13], [14]. CI has been widely used in decentralized sensor network data fusion algorithms (see e.g., [15]–[21]). However, CI often results in highly conservative estimates with covariance much larger than the actual one. Therefore, alternative approaches that strive for fusion results with a smaller error covariance matrix have been proposed in the literature. For example, the Largest Ellipsoid method of [22] employs geometrical transformations to find the largest ellipsoid inscribed within the intersection region of the two local estimates. But, it does not provide an efficient method to compute the mean of the fused track. Ellipsoidal Intersection (EI) method of [23] intends to reduce the conservatism of CI method by modifying its equations to include parameters that take into account the maximum possible common information shared by the tracks. However, the parameters introduced in the EI method are not sufficient to guarantee consistency. In a recent work, [24] has modified the parameters of the EI method to devise a consistent fusion algorithm called the Inverse Covariance Intersection (ICI) method.

In this paper, we revisit the problem of track-to-track fusion in the absence of cross-covariance information. Due to the importance of estimation accuracy in sensor networks, we propose a novel fusion method that trades in extra computation for a better fusion performance. In our method, instead of conservatively over-bounding the joint covariance matrix of the tracks, we aim to construct the unknown cross-covariance matrix. We start by observing that the cross-covariance matrix of correlated tracks can be stated as the product of the square root of the covariance matrices of the tracks and a contraction matrix. We then propose an appropriate optimization problem to obtain an estimate for this contraction matrix in a way that the fused track is less conservative than the one obtained by the CI method. Moreover, we show that our fused track, as

The first and the second authors are with Aerospace Engineering Department, Sharif University of Technology, Tehran, Iran e-mail: zareijalalabadi_m,malaek@ae.sharif.edu.

Solmaz S. Kia is with Mechanical and Aerospace Engineering Department, University of California Irvine, Irvine, CA 92697, e-mail: solmaz@uci.edu.

expected from an approximation technique, is conservative than the optimal track that one can obtain by taking into account the exact value of the cross-covariance between the tracks. We validate our fusion algorithm's properties through rigorous analysis. Furthermore, we show that our design optimization problem can be cast as a difference of convex (DC) programming problem [25], which can be solved in an efficient manner using DC programming software solutions [26]. Finally, we show parallels between the structure we use to estimate the cross-covariance matrix and the estimated cross-covariance that one can calculate using sigma points generated via Unscented Kalman Filter (UKF) type [27] approach. We compare the performance of our algorithm to the CI and the ICI methods in a numerical simulation.

Notations: $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ are, respectively, the set of real positive semi definite and positive definite matrices. For $\mathbf{M}\in\mathbb{S}^n_+$, its matrix square root is $\sqrt{\mathbf{M}}$, i.e., $\sqrt{\mathbf{M}}^\top\sqrt{\mathbf{M}}=\mathbf{M}$. A function $f:\mathbb{R}^n\to\mathbb{R}$ is a difference of convex function iff there exist convex functions $g,h:\mathbb{R}^n\to\mathbb{R}$ such that f can be decomposed as the difference between g and h, i.e., $f(\mathbf{x})=g(\mathbf{x})-h(\mathbf{x})$ for $\mathbf{x}\in\mathbb{R}^n$ (c.f. [25]). A matrix $\mathbf{C}\in\mathbb{R}^{n\times m}$ is called a *contraction matrix* if its spectral norm satisfies $\|\mathbf{C}\|_2\leq 1$; it is a *strict contraction* if $\|\mathbf{C}\|_2<1$. The set of $n\times m$ real strict contraction matrices is denoted by $\mathcal{C}^{n\times m}$.

Lemma 1.1 (c.f. [28, page 207 and page 350]): Let $\mathbf{A}\in\mathbb{R}^{n\times n}$ and $\mathbf{B}\in\mathbb{R}^{m\times m}$, and $\mathbf{X}\in\mathbb{R}^{m\times n}$ be given. Then, the joint matrix $\left[\begin{smallmatrix}\mathbf{A} & \mathbf{X}\\ \mathbf{X}^\top & \mathbf{B}\end{smallmatrix}\right]$ is positive semi definite (positive definite) if and only if $\mathbf{A}$ and $\mathbf{B}$ are positive semi definite (positive definite) and there is a contraction (strict contraction) matrix $\mathbf{C}$ such that $\mathbf{X}=\sqrt{\mathbf{A}}^\top\,\mathbf{C}\,\sqrt{\mathbf{B}}$.

II. Problem definition and objective statement

Consider sensor stations s_i and s_j each with processing and transmission capabilities. These stations at each time k observe and compute a local estimate and its corresponding error covariance ($\hat{\mathbf{x}}_r(k)\in\mathbb{R}^n$, $\mathbf{P}_r(k)\in\mathbb{S}^n_{++}$), $r\in\{i,j\}$, of a time-varying target/process whose true state is denoted by $\mathbf{x}(k)\in\mathbb{R}^n$. We assume that the tracks are synchronized. The tracks are transmitted to a fusion center so that a better estimate is generated by fusing them (in distributed estimation problems, the tracks are transmitted to the neighboring agents). To simplify the notation, in the following we drop the time index k of the estimates. Since the tracks from stations s_i and s_j are originated from observing a common system whose process noise is the same for both stations, these tracks are correlated [6]. This means that the cross-covariance $\mathbf{P}_{ij}$ between the tracks is nonzero. The tracks are normally fused by linearly combining them to obtain an improved estimate ($\hat{\mathbf{x}}_c\in\mathbb{R}^n$, $\mathbf{P}_c\in\mathbb{S}^n_{++}$), i.e., $\hat{\mathbf{x}}_c=\mathbf{W}_i\hat{\mathbf{x}}_i+\mathbf{W}_j\hat{\mathbf{x}}_j$, with error covariance

$$\begin{aligned}\mathbf{P}_c =&E\big[(\mathbf{x}-\hat{\mathbf{x}}_c)\,(\mathbf{x}-\hat{\mathbf{x}}_c)^\top\big]=\mathbf{W}_i\mathbf{P}_i\mathbf{W}_i^\top+\mathbf{W}_j\mathbf{P}_j\mathbf{W}_j^\top\\ &+\mathbf{W}_i\mathbf{P}_{ij}\mathbf{W}_j^\top+\mathbf{W}_j\mathbf{P}_{ij}^\top\mathbf{W}_i^\top=\mathbf{W}\,\mathbf{P}_J\,\mathbf{W}^\top,\end{aligned}\qquad(1)$$

where $\mathbf{W}\triangleq\big[\mathbf{W}_i\ \mathbf{W}_j\big]$ and $\mathbf{P}_J\triangleq\left[\begin{smallmatrix}\mathbf{P}_i & \mathbf{P}_{ij}\\ \mathbf{P}_{ij}^\top & \mathbf{P}_j\end{smallmatrix}\right]$ is the joint covariance matrix. When the cross-covariance $\mathbf{P}_{ij}$ is known, the consistent minimum variance fused estimate is given as follows.

Theorem 2.1 (c.f. [6], [13]): given ($\hat{\mathbf{x}}_r$, $\mathbf{P}_r$), $r\in\{i,j\}$, and $\mathbf{P}_{ij}$, the solution of

$$\mathbf{W}=\operatorname{argmin}\det\big(\mathbf{P}_c\big)\quad\text{subject to}\quad\mathbf{W}\left[\begin{smallmatrix}\mathbf{I}_n\\ \mathbf{I}_n\end{smallmatrix}\right]=\mathbf{I}_n,\qquad(2)$$

yields a consistent estimate $\hat{\mathbf{x}}_c^\star$ with covariance $\mathbf{P}_c^\star$ as follows:

$$\hat{\mathbf{x}}_c^\star=\hat{\mathbf{x}}_i+(\mathbf{P}_i-\mathbf{P}_{ij})(\mathbf{P}_i+\mathbf{P}_j-\mathbf{P}_{ij}-\mathbf{P}_{ij}^\top)^{-1}(\hat{\mathbf{x}}_j-\hat{\mathbf{x}}_i),\qquad(3a)$$

$$\mathbf{P}_c^\star=\mathbf{P}_i-(\mathbf{P}_i-\mathbf{P}_{ij})(\mathbf{P}_i+\mathbf{P}_j-\mathbf{P}_{ij}-\mathbf{P}_{ij}^\top)^{-1}(\mathbf{P}_i-\mathbf{P}_{ij}^\top).\qquad(3b)$$

□

The minimum variance fused estimate (3) requires exact knowledge of the cross-covariance matrix $\mathbf{P}_{ij}$. The *cross-covariance propagation method for track-to-track fusion* (c.f. [29]) for a linear process with state $\mathbf{x}$ and measurements $\mathbf{z}_r$

$$\begin{aligned}&\mathbf{x}(k+1)=\mathbf{F}(k)\mathbf{x}(k)+\boldsymbol{\omega}(k),\quad\boldsymbol{\omega}(k)\sim\mathcal{N}(\mathbf{0}_{n\times1},\mathbf{Q}(k))\\ &\mathbf{z}_r(k)=\mathbf{H}_r(k)\mathbf{x}(k)+\boldsymbol{\nu}_r(k),\quad\boldsymbol{\nu}(k)\sim\mathcal{N}(\mathbf{0}_{n\times1},\mathbf{R}_r(k))\\ &r=\{i,j\},\ E[\boldsymbol{\nu}_i(k)\boldsymbol{\nu}_j^\top(k)]=\mathbf{0}_n\end{aligned}$$

gives the exact cross-covariance matrix $\mathbf{P}_{ij}(k)$ as

$$\begin{aligned}\mathbf{P}_{ij}(k)=&(\mathbf{I}_n-\mathbf{K}_i(k)\mathbf{H}_i(k))\big(\mathbf{F}(k-1)\mathbf{P}_{ij}(k-1)\mathbf{F}^\top(k-1)\\ &+\mathbf{Q}(k-1)\big)(\mathbf{I}_n-\mathbf{K}_j(k)\mathbf{H}_j(k))^\top.\end{aligned}\qquad(4)$$

Assuming that the initial measurements are uncorrelated, the initial condition of (4) is set to zero. Although equation (4) computes the exact cross-covariance $\mathbf{P}_{ij}$, the method is restricted to applications that sensors implement the Kalman filter. Also, note that to maintain $\mathbf{P}_{ij}$ using (4), the Kalman gains $\mathbf{K}_i(k)$ and $\mathbf{K}_j(k)$ of the sensor stations have to be transmitted to the fusion center. Therefore, the cross-covariance propagation method requires communication between sensor stations or sensor stations to fusion center at all time steps k. This requirement incurs a high communication cost on the sensor stations and is vulnerable to communication failure.

CI method, as mentioned earlier, is a popular track-to-track fusion algorithm that does not require an explicit knowledge of the cross-covariance of the sensors' tracks. For tracks $(\hat{\mathbf{x}}_r,\mathbf{P}_r)$, $r\in\{i,j\}$ with unknown correlation, CI fused track is

$$\mathbf{P}_{\text{CI}}^{-1}=\omega\,\mathbf{P}_i^{-1}+(1-\omega)\,\mathbf{P}_j^{-1},\qquad(5a)$$

$$\hat{\mathbf{x}}_{\text{CI}}=\mathbf{P}_{\text{CI}}\Big(\omega\mathbf{P}_i^{-1}\hat{\mathbf{x}}_i+(1-\omega)\mathbf{P}_j^{-1}\hat{\mathbf{x}}_j\Big),\qquad(5b)$$

where $0\leq\omega\leq1$ is a weighting factor that can be used to optimize the updates with respect to a performance criterion, such as minimizing the trace or the determinant of $\mathbf{P}_{\text{CI}}$, e.g.,

$$\omega^\star=\operatorname{argmin}\ \det\big(\mathbf{P}_{\text{CI}}\big),\quad\text{subject to}\quad 0\leq\omega\leq1.\qquad(6)$$

Given (5a), the optimization algorithm (6) produces a fused estimate that satisfies $\det(\mathbf{P}_{\text{CI}})\leq\min\big(\det(\mathbf{P}_i),\det(\mathbf{P}_j)\big)$. Moreover, for consistent estimates $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$, the fused estimate $\hat{\mathbf{x}}_{\text{CI}}$ is guaranteed to be consistent [30]. However, the CI fusion is a suboptimal fusion method compared to the method (3), indeed, $\mathbf{P}_{\text{CI}} \geq \mathbf{P}_c$ for all $\mathbf{P}_c$ satisfying (1). More precisely, as shown in [29], the error ellipsoid corresponding to $\mathbf{P}_c$ is located inside the intersection of the error ellipsoids corresponding to $\mathbf{P}_i$ and $\mathbf{P}_j$ while the error ellipsoid corresponding to $\mathbf{P}_{\text{CI}}$ circumscribes the intersection of the error ellipsoids corresponding to $\mathbf{P}_i$ and $\mathbf{P}_j$, see Fig. 1 for some numerical examples.

Our aim is to devise an alternative track-to-track fusion algorithm which analogous to the CI method does not require the exact knowledge of correlation information between the tracks, however, it produces a fused track that is less conservative.

III. Maximum Allocated Covariance

In this section, we present a novel track-to-track fusion algorithm, called Maximum Allocated Covariance (MAC) method, for tracks with unknown correlations. The design of MAC is based on the construction of the cross-covariance matrix in a way that the determinant of the fused covariance matrix acquires the maximum allocatable value. In doing so, we guarantee that MAC's estimate is always conservative than the estimate one obtains when cross-covariance matrix is known. In the result below, we show also that MAC out-preforms the CI method by producing an estimate whose covariance matrix is smaller than the one CI provides.

Theorem 3.1 (MAC track-to-track fusion): Consider two unbiased and consistent tracks $(\hat{\mathbf{x}}_r \in \mathbb{R}^n, \mathbf{P}_r \in \mathbb{S}^n_{++})$, $r \in \{i,j\}$, with an unknown correlation. Let

$$\mathbf{F}(\mathbf{X}) = \mathbf{P}_i - (\mathbf{P}_i - \mathbf{X})\big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X} - \mathbf{X}^\top\big)^{-1}(\mathbf{P}_i - \mathbf{X}^\top), \quad (7)$$

and $\mathbf{X}^\star = \sqrt{\mathbf{P}_i}^\top \mathbf{C}^\star \sqrt{\mathbf{P}_j}$ where

$$\mathbf{C}^\star = \underset{\mathbf{C} \in \mathcal{C}^{n\times n}}{\operatorname{argmax}} \det\big(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top \mathbf{C} \sqrt{\mathbf{P}_j})\big). \quad (8)$$

Then, the fused track $(\hat{\mathbf{x}}_{\text{MAC}}\ , \mathbf{P}_{\text{MAC}})$ given by $\hat{\mathbf{x}}_{\text{MAC}} = \mathbf{W}_i \hat{\mathbf{x}}_i + \mathbf{W}_j \hat{\mathbf{x}}_j$ and $\mathbf{P}_{\text{MAC}} = \mathbf{F}(\mathbf{X}^\star)$, with

$$\begin{aligned} \mathbf{W}_i &= (\mathbf{P}_j - \mathbf{X}^{\star\top})\big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\big)^{-1}, \\ \mathbf{W}_j &= (\mathbf{P}_i - \mathbf{X}^\star)\big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\big)^{-1}, \end{aligned} \quad (9)$$

satisfies

$$E[\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}}] = \mathbf{0}_{n\times 1}, \quad (10a)$$

$$\mathbf{0}_n \leq \mathbf{P}_{\text{MAC}} \leq \mathbf{P}_r\ , r \in \{i,j\}, \quad (10b)$$

$$\mathbf{0}_n \leq \mathbf{P}_{\text{MAC}} \leq \mathbf{P}_{\text{CI}}, \quad 0 \leq \omega \leq 1, \quad (10c)$$

$$\det(\mathbf{P}_c^\star) \leq \det(\mathbf{P}_{\text{MAC}}) \leq \det(\mathbf{P}_{\text{CI}}), \quad 0 \leq \omega \leq 1, \quad (10d)$$

where $\mathbf{P}_c^\star$ and $\mathbf{P}_{\text{CI}}$ are given by (3b) and (5a), respectively.

Proof: (10a) is satisfied, because the tracks from sensor stations s_i and s_j are unbiased and also the gains in (9) satisfy $\mathbf{W}_i + \mathbf{W}_j = \mathbf{I}_n$ resulting in $E[\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}}] = E[(\mathbf{W}_i + \mathbf{W}_j)\mathbf{x} - (\mathbf{W}_i \hat{\mathbf{x}}_i + \mathbf{W}_j \hat{\mathbf{x}}_j)] = \mathbf{W}_i E[\mathbf{x} - \hat{\mathbf{x}}_i] + \mathbf{W}_j E[\mathbf{x} - \hat{\mathbf{x}}_j] = \mathbf{0}_{n\times 1} + \mathbf{0}_{n\times 1} = \mathbf{0}_{n\times 1}$. Next, we show the validity of statement (10b). Let $\mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j) = \Big\{\mathbf{X} \in \mathbb{R}^{n\times n} \Big| \Big[\begin{smallmatrix} \mathbf{P}_i & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{P}_j \end{smallmatrix}\Big] \in \mathbb{S}^{2n}_{++}\Big\}$. For any $\mathbf{X} \in \mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j)$, using a congruent transformation with non-singular transformation matrix $\mathbf{T} = \big[\begin{smallmatrix} \mathbf{I}_n & \mathbf{I}_n \\ \mathbf{0}_n & -\mathbf{I}_n \end{smallmatrix}\big]$ we obtain

$$\bar{\mathbf{P}} = \mathbf{T}^\top \begin{bmatrix} \mathbf{P}_i & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{P}_j \end{bmatrix} \mathbf{T} = \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_i - \mathbf{X} \\ \mathbf{P}_i - \mathbf{X}^\top & \mathbf{P}_i + \mathbf{P}_j - \mathbf{X} - \mathbf{X}^\top \end{bmatrix} > \mathbf{0}_{2n}, \quad (11)$$

which by virtue of the Schur complement (c.f. [31, page 495]) of a block positive definite matrix we have the guarantees that

$$\mathbf{P}_i + \mathbf{P}_j - \mathbf{X} - \mathbf{X}^\top > \mathbf{0}_n, \quad (12a)$$

$$\mathbf{F}(\mathbf{X}) = \mathbf{P}_i - (\mathbf{P}_i - \mathbf{X})\big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X} - \mathbf{X}^\top\big)^{-1}(\mathbf{P}_i - \mathbf{X}^\top) > \mathbf{0}_n. \quad (12b)$$

By virtue of Lemma 1.1, any $\mathbf{X} \in \mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j)$ can be written as $\mathbf{X} = \sqrt{\mathbf{P}_i}^\top \mathbf{C} \sqrt{\mathbf{P}_j}$ for a $\mathbf{C} \in \mathcal{C}^{n\times n}$. Therefore, we have $\mathbf{X}^\star \in \mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j)$, and as a result $\mathbf{P}_{\text{MAC}} = \mathbf{F}(\mathbf{X}^\star) > \mathbf{0}_n$ follows from (12b). Subsequently, from (12a) we have $(\mathbf{P}_i - \mathbf{X}^\star)\big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\big)^{-1}(\mathbf{P}_i - \mathbf{X}^{\star\top}) \geq \mathbf{0}_n$, which gives us $\mathbf{P}_i - \mathbf{P}_{\text{MAC}} = (\mathbf{P}_i - \mathbf{X}^\star)(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top})^{-1}(\mathbf{P}_i - \mathbf{X}^{\star\top}) \geq \mathbf{0}_n$, or equivalently $\mathbf{P}_{\text{MAC}} \leq \mathbf{P}_i$. To complete our proof of statement (10b), remark that we can also write $\mathbf{P}_{\text{MAC}}$ as $\mathbf{P}_{\text{MAC}} = \mathbf{P}_j - (\mathbf{P}_j - \mathbf{X}^{\star\top})\big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\big)^{-1}(\mathbf{P}_j - \mathbf{X}^\star)$.

Therefore, following the similar steps as above but with congruent transformation matrix $\big[\begin{smallmatrix} -\mathbf{I}_n & \mathbf{0}_n \\ \mathbf{I}_n & \mathbf{I}_n \end{smallmatrix}\big]$, we can show that $\mathbf{P}_{\text{MAC}} - \mathbf{P}_j \leq \mathbf{0}_n$ or $\mathbf{P}_{\text{MAC}} \leq \mathbf{P}_j$.

Next, we show (10c) and (10d) hold. Since $\mathbf{P}_{\text{MAC}} \leq \mathbf{P}_r$, $r \in \{i,j\}$, we have $\mathbf{P}_r^{-1} \leq \mathbf{P}_{\text{MAC}}^{-1}$, $r \in \{i,j\}$. Therefore, for $0 \leq \omega \leq 1$, we can write $\mathbf{P}_{\text{CI}}^{-1} = \omega\, \mathbf{P}_i^{-1} + (1-\omega)\mathbf{P}_j^{-1} \leq \omega\, \mathbf{P}_{\text{MAC}}^{-1} + (1-\omega)\mathbf{P}_{\text{MAC}}^{-1} = \mathbf{P}_{\text{MAC}}^{-1}$. As a result, $\mathbf{P}_{\text{MAC}} \leq \mathbf{P}_{\text{CI}}$ for any $0 \leq \omega \leq 1$. Subsequently, we conclude also that $\det(\mathbf{P}_{\text{MAC}}) \leq \det(\mathbf{P}_{\text{CI}})$ for all $0 \leq \omega \leq 1$. To prove $\det(\mathbf{P}_c^\star) \leq \det(\mathbf{P}_{\text{MAC}})$, notice that the true cross-covariance matrix satisfies $\mathbf{P}_{ij} \in \mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j)$. Therefore, there exists $\mathbf{C}_c \in \mathcal{C}^{n\times n}$ such that $\mathbf{P}_{ij} = \sqrt{\mathbf{P}_i}^\top \mathbf{C}_c \sqrt{\mathbf{P}_j}$. Since $\mathbf{P}_c^\star = \mathbf{F}(\mathbf{P}_{ij})$, then from (8) we have that $\det(\mathbf{P}_c^\star) \leq \det(\mathbf{P}_{\text{MAC}})$. ■

(10a) ensures that $\hat{\mathbf{x}}_{\text{MAC}}$ is unbiased, while (10b) guarantees that MAC fusion produces a better estimate than the individual tracks of the sensor stations s_i and s_j. Because of (10c), MAC also guarantees that its fused track is less conservative than the track obtained from the CI fusion regardless of the value of $\omega \in [0,1]$. The lower bound in (10d) ensures that the fused track is not optimistic in comparison to the optimal fused track, while the upper bound is an obvious outcome of (10c). Here, we used $\det(\mathbf{P})$ as the scalar measure to compare the total uncertainty of MAC fusion to others. The relationships described in (10b), (10c), and (10d) are evident in Fig. 1 for three numerical examples.

Our next result shows that the optimization problem (8) can be cast in an equivalent DC programming optimization form, enabling us to obtain MAC track-to-track fusion in an efficient manner using DC programming software solutions.

Theorem 3.2 (MAC is a DC programming problem): The optimization problem (8) as defined in Theorem 3.1 is equivalent to the DC programming problem below

$$\mathbf{C}^\star = \underset{\mathbf{C}\in\mathcal{C}^{n\times n}}{\operatorname{argmin}} \Big(-\operatorname{Log}\det(\begin{bmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{bmatrix}) + \operatorname{Log}\big(\det(\mathbf{P}_i+\mathbf{P}_j-\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}-\sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i}))\Big). \tag{13}$$

Proof: Recall that for $\mathbf{C}\in\mathcal{C}^{n\times n}$ by definition we have $\left[\begin{smallmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{smallmatrix}\right] > \mathbf{0}_n$. Also we already showed that for $\mathbf{C}\in\mathcal{C}^{n\times n}$ we have $(\mathbf{P}_i+\mathbf{P}_j - \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j} - \sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i}) > \mathbf{0}_n$. Since Log of determinant of positive definite matrices is a concave function (c.f. [31, page 488]) and negative of a concave function is convex, the optimization problem (13) is a DC programming problem. Next, we show that optimization problem (8) is equivalent to (13). For $\mathbf{X}\in\mathcal{J}_{++}(\mathbf{P}_i\ ,\mathbf{P}_j)$, from the proof of Theorem 3.1 recall that matrix $\overline{\mathbf{P}}$ in (11) is positive definite and its Schur complement components are (12). Using Schur complement and its determinant formula (c.f. [31, page 24]), $\det(\overline{\mathbf{P}})\neq 0$ can be written as $\det(\overline{\mathbf{P}}) = \det(\mathbf{F}(\mathbf{X}))\det(\mathbf{P}_i+\mathbf{P}_j-\mathbf{X}-\mathbf{X}^\top)$. Therefore,

$$\det(\mathbf{F}(\mathbf{X})) = \det(\overline{\mathbf{P}})\det(\mathbf{P}_i+\mathbf{P}_j-\mathbf{X}-\mathbf{X}^\top)^{-1}. \tag{14}$$

Note that $\det(\overline{\mathbf{P}}) = \det(\mathbf{T}^\top)\det(\left[\begin{smallmatrix}\mathbf{P}_i & \mathbf{X}\\ \mathbf{X}^\top & \mathbf{P}_j\end{smallmatrix}\right])\det(\mathbf{T}) = \det(\left[\begin{smallmatrix}\mathbf{P}_i & \mathbf{X}\\ \mathbf{X}^\top & \mathbf{P}_j\end{smallmatrix}\right])$. Recall that any $\mathbf{X}\in\mathcal{J}_{++}(\mathbf{P}_i\ ,\mathbf{P}_j)$ can be written as $\mathbf{X} = \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}$ where $\mathbf{C}\in\mathcal{C}^{n\times n}$. Because $\left[\begin{smallmatrix}\mathbf{P}_i & \mathbf{X}\\ \mathbf{X}^\top & \mathbf{P}_j\end{smallmatrix}\right] = \left[\begin{smallmatrix}\sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_i} & \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}\\ \sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i} & \sqrt{\mathbf{P}_j}^\top\sqrt{\mathbf{P}_j}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\sqrt{\mathbf{P}_i}^\top & \mathbf{0}_n\\ \mathbf{0}_n & \sqrt{\mathbf{P}_j}^\top\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{smallmatrix}\right]\left[\begin{smallmatrix}\sqrt{\mathbf{P}_i} & \mathbf{0}_n\\ \mathbf{0}_n & \sqrt{\mathbf{P}_j}\end{smallmatrix}\right]$, we can write $\det(\overline{\mathbf{P}}) = \det(\sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_i})\det(\sqrt{\mathbf{P}_j}^\top\sqrt{\mathbf{P}_j})\det(\left[\begin{smallmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{smallmatrix}\right])$. As a result, for $\mathbf{C}\in\mathcal{C}^{n\times n}$ from (14), we obtain

$$\operatorname{Log}(\det(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}))) = \operatorname{Log}(\det(\mathbf{P}_i))+ \operatorname{Log}(\det(\mathbf{P}_j)) + \operatorname{Log}(\det(\left[\begin{smallmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{smallmatrix}\right])) - \operatorname{Log}(\det(\mathbf{P}_i+\mathbf{P}_j-\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}-\sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i})). \tag{15}$$

Recall that $\left[\begin{smallmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{smallmatrix}\right] > \mathbf{0}_n$ and $(\mathbf{P}_i+\mathbf{P}_j-\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}-\sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i}) > \mathbf{0}_n$. As a result, since Log is a strictly increasing function over positive real numbers, we can write the maximization problem (8) in the equivalent DC programming minimization problem (13). This completes our proof. ■

Next, we point out an interesting relationship between the estimated cross-covariance matrix that MAC generates and the estimate that one can obtain by generating sigma points around the mean and applying a method similar to UKF [27]. Following the UKF approach, $2n+1$ sigma points for the sensor stations s_r, $r\in\{i,j\}$ are given as $\mathbf{x}_r^{(0)} = \hat{\mathbf{x}}_r$, $\mathbf{x}_r^{(\ell)} = \hat{\mathbf{x}}_r + \big(\sqrt{(n+\lambda)\mathbf{P}_r}\big)_\ell^\top$, and $\mathbf{x}_r^{(\ell+n)} = \hat{\mathbf{x}}_r - \big(\sqrt{(n+\lambda)\mathbf{P}_r}\big)_\ell^\top$ for $\ell\in\{1,\dots,n\}$, where $\big(\sqrt{(n+\lambda)\mathbf{P}_r}\big)_\ell^\top$ is the ℓ-th row of matrix square root of $(n+\lambda)\mathbf{P}_r$. The weighting coefficients are defined as $w^{(0)} = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta$ and $w^{(\ell)} = \frac{1}{2(n+\lambda)}$, for $\ell = 1,\dots,2n$, where α, β, and λ are constant tuning parameters. The cross-covariance of two local estimates at stations s_i and s_j then can be calculated as $\mathbf{P}_{ij} = \sum_{\ell=1}^{2n} w^{(\ell)} E\big[\big(\hat{\mathbf{x}}_i - \mathbf{x}_i^{(\ell)}\big)\big(\hat{\mathbf{x}}_j - \mathbf{x}_j^{(\ell)}\big)^\top\big] = 2\sum_{\ell=1}^{n} w^{(\ell)} E\big[\big(\hat{\mathbf{x}}_i - \mathbf{x}_i^{(\ell)}\big)\big(\hat{\mathbf{x}}_j - \mathbf{x}_j^{(\ell)}\big)^\top\big]$, (here we used $\mathbf{x}_r^{(\ell)} = -\mathbf{x}_r^{(\ell+n)}$, $r\in\{i,j\}$). Given the definition of the sigma points, then, the estimated cross-covariance by sigma points is $\mathbf{P}_{ij} = \sum_{\ell=1}^{n}\big(\sqrt{\mathbf{P}_i}\big)_\ell^\top\big(\sqrt{\mathbf{P}_j}\big)_\ell = \sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_j}$. This cross-covariance guarantees the positive semi-definiteness of joint covariance matrix but it over-estimates the estimates' correlation.

We close this section with few remarks regarding the consistency analysis of MAC. Given a process with random state $\mathbf{x}$, a state estimator which produces an estimate $\hat{\mathbf{x}}$ with the associated error covariance $\mathbf{P}$ is said to be consistent if it is unbiased, i.e., $E[\mathbf{x}-\hat{\mathbf{x}}] = \mathbf{0}_{n\times 1}$, and its estimates satisfy $\mathbf{P} = E[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top]$ (see e.g., [29], [32], [33]). The definition of estimator consistency sometimes is relaxed from covariance matching to $\mathbf{P} \geq E[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top]$ (see e.g., [13]). In Theorem 3.1, we have shown that when the local tracks are unbiased, MAC generates an unbiased fused track. Next, we examine the covariance consistency of the MAC. Given MAC's fusion gains in (9), with appropriate manipulations, we can write $\mathbf{P}_{\text{MAC}} = \mathbf{W}_i\mathbf{P}_i\mathbf{W}_i^\top + \mathbf{W}_i\mathbf{X}^\star\mathbf{W}_j^\top + \mathbf{W}_j\mathbf{X}^{\star\top}\mathbf{W}_i^\top + \mathbf{W}_j\mathbf{P}_j\mathbf{W}_j^\top$. On the other hand, using the same gains, we have (recall $\mathbf{W}_i + \mathbf{W}_j = \mathbf{I}_n$)

$$\begin{aligned} &E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top] = \\ &E\big[\big(\mathbf{W}_i(\mathbf{x}-\hat{\mathbf{x}}_i)+\mathbf{W}_j(\mathbf{x}-\hat{\mathbf{x}}_j)\big)\big(\mathbf{W}_i(\mathbf{x}-\hat{\mathbf{x}}_i)+\mathbf{W}_j(\mathbf{x}-\hat{\mathbf{x}}_j)\big)^\top\big] \\ &= \mathbf{W}_i\mathbf{P}_i\mathbf{W}_i^\top + \mathbf{W}_i\mathbf{P}_{ij}\mathbf{W}_j^\top + \mathbf{W}_j\mathbf{P}_{ij}^\top\mathbf{W}_i^\top + \mathbf{W}_j\mathbf{P}_j\mathbf{W}_j^\top. \end{aligned} \tag{16}$$

As a result, we can write $\mathbf{P}_{\text{MAC}} - E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top] = \mathbf{W}\,\mathbf{M}\,\mathbf{W}^\top$, where $\mathbf{W} \triangleq \left[\begin{smallmatrix}\mathbf{W}_i & \mathbf{W}_j\end{smallmatrix}\right]$, and $\mathbf{M} \triangleq \left[\begin{smallmatrix}\mathbf{0}_n & \mathbf{X}^\star-\mathbf{P}_{ij}\\ (\mathbf{X}^\star-\mathbf{P}_{ij})^\top & \mathbf{0}_n\end{smallmatrix}\right]$. For $\mathbf{X}^\star\neq\mathbf{P}_{ij}$, matrix $\mathbf{M}$ is an indefinite matrix (neither positive (semi-) definite nor negative (semi-) definite) whose eigenvalues are real and symmetric around origin (c.f. [34]). Although matrix $\mathbf{M}$ is indefinite, the product matrix $\mathbf{W}\,\mathbf{M}\,\mathbf{W}^\top$ is not necessarily indefinite, i.e., depending on the numerical values of $\mathbf{W}$ and $\mathbf{X}^\star - \mathbf{P}_{ij}$, $\mathbf{P}_{\text{MAC}} - E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top] = \mathbf{W}\,\mathbf{M}\,\mathbf{W}^\top$ can be positive/negative (semi-) definite or indefinite. For example, the table in Fig. 1 lists $\mathbf{P}_{\text{MAC}}$ and $E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$ for three numerical examples. Fig. 1 also depicts the corresponding 1σ error ellipses of these examples. As shown, in the cases (a) and (b), MAC produces consistent estimates but in case (c) the covariance consistency condition is slightly violated.

Consistency of state estimation filters is measured also by statistical consistency tests such as the Average Normalized Estimation Error Squared (ANEES) [33]. These tests are based on Monte Carlo simulations which provide M independent samples of estimation error $\mathbf{e}_\ell(k) = \mathbf{x}_\ell(k) - \hat{\mathbf{x}}_\ell(k)$,

Case	$\mathbf{P}_i$	$\mathbf{P}_j$	$\mathbf{P}_{ij}$	$\mathbf{P}_{\text{MAC}}$	$E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\ldots)^\top]$
(a)	$\begin{bmatrix}20 & 0\\ 0 & 9\end{bmatrix}$	$\begin{bmatrix}10 & 0\\ 0 & 18\end{bmatrix}$	$\begin{bmatrix}5.66 & 0.95\\ 0.47 & -6.36\end{bmatrix}$	$\begin{bmatrix}9.96 & 0.29\\ 0.29 & 8.96\end{bmatrix}$	$\begin{bmatrix}9.6 & -0.12\\ -0.12 & 7.43\end{bmatrix}$
(b)	$\begin{bmatrix}20 & 6.5\\ 6.5 & 9\end{bmatrix}$	$\begin{bmatrix}10 & 3.3\\ 3.3 & 18\end{bmatrix}$	$\begin{bmatrix}5.66 & 1.87\\ 1.84 & -4.79\end{bmatrix}$	$\begin{bmatrix}10.00 & 3.26\\ 3.26 & 7.95\end{bmatrix}$	$\begin{bmatrix}10.00 & 3.26\\ 3.26 & 7.95\end{bmatrix}$
(c)	$\begin{bmatrix}20 & 6\\ 6 & 9\end{bmatrix}$	$\begin{bmatrix}10 & 4\\ 4 & 18\end{bmatrix}$	$\begin{bmatrix}5.66 & 3.17\\ 2.12 & -4.31\end{bmatrix}$	$\begin{bmatrix}9.94 & 3.26\\ 3.26 & 8.26\end{bmatrix}$	$\begin{bmatrix}10.09 & 4.29\\ 4.29 & 8.81\end{bmatrix}$

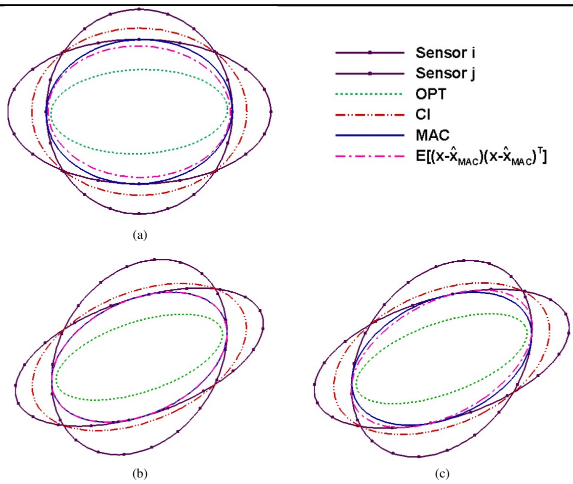


Fig. 1: 1σ ellipses for the fused error covariance matrix computed by MAC, CI and OPT (the optimal filter (3)) fusion methods. In comparison to OPT, MAC and CI fusions are both conservative, but MAC algorithm produces better results. The plots also depict how $\mathbf{P}_{\text{MAC}}$ compares to $E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$ given in (16): plot (a) demonstrates an example in which $\mathbf{P}_{\text{MAC}}$ is larger than $E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$; plot (b) demonstrates a case where $\mathbf{P}_{\text{MAC}}$ almost matches $E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$, and plot (c) demonstrates a case that $\mathbf{P}_{\text{MAC}}$ intersects $E[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$.

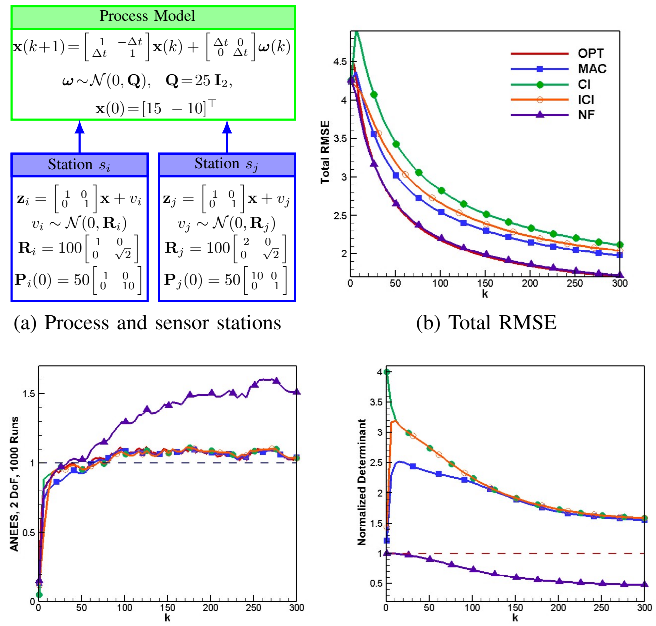
where $\mathbf{x}_\ell(k) \in \mathbb{R}^n$ is the true state and $\hat{\mathbf{x}}_\ell(k) \in \mathbb{R}^n$ is the filter estimate with the associated error covariance $\mathbf{P}_\ell(k) \in \mathbb{S}^n_{++}$ at the Monte Carlo run $\ell \in \{1, \cdots, M\}$. In the ANEES test, the consistency measure is

$$\bar{\epsilon}(k) = \frac{1}{nM}\sum\nolimits_{\ell=1}^{M} \mathbf{e}_\ell^\top(k)\mathbf{P}_\ell^{-1}(k)\mathbf{e}_\ell(k).$$

An estimator is consistent, if ANEES measure $\bar{\epsilon}(k)$ converges to 1. If ANEES measure $\bar{\epsilon}(k) \gg 1$, the estimator is optimistic which implies the actual estimation error is much larger than what the estimator believes, while $\bar{\epsilon}(k) \ll 1$ indicates the actual estimation error is much smaller than what the estimator believes i.e., the estimator is too pessimistic [35]. A simulation study demonstrating the ANEES measure for MAC fusion is given in Section IV, see Fig. 2(c).

IV. Simulation

We demonstrate the performance of the MAC track-to-track fusion algorithm in a simulation study for two sensor stations. The process and sensor measurement models of these stations, taken from [36], are described in Fig. 2(a). Each sensor station uses a local Kalman filter to generate its local track at a



(a) Process and sensor stations

(b) Total RMSE

(c) ANEES

(d) Normalized Determinant

Fig. 2: A comparison study: OPT fusion rate is $\frac{1}{\Delta t} = 50$Hz, while MAC, CI, ICI, and NF methods' fusion rate is 10Hz.

rate $\frac{1}{\Delta t} = 50$Hz. We use an optimal fusion algorithm (OPT fusion) consisted of the optimal track-to-track fusion (3) and the cross-covariance propagation (4) to create the optimal reference fusion results. For the OPT fusion, the fusion center needs to communicate with the sensor stations at 50Hz rate. We also implement MAC, CI, and a Naive Fusion (NF) methods to perform fusion at 10Hz. NF is a track-to-track fusion method which uses (3) with $\mathbf{P}_{ij} = \mathbf{0}_n$, i.e., it disregards the correlation between the tracks. We also compare MAC to the ICI method of [24]. Our simulation results in Fig. 2 are generated from 1000 Monte Carlo runs. Fig. 2(b) demonstrates the total RMSE plot of the fusion methods. As expected, OPT has the best performance. We can see also that MAC and ICI preform better than CI. NF appears to preform better than CI, ICI and MAC but as we can see in the ANEES test results (see Fig. 2(c)) this method is not consistent. Whereas, MAC's ANEES is comparable with OPT and CI, demonstrating a consistent behavior with an ANEES measure near 1. Fig. 2(d) shows the averaged (over Monte Carol runs) of the normalized (with respect to OPT fusion results) total uncertainty of the fused tracks. The total uncertainty is measured in terms of the determinant of the covariance of the fused tracks. This plot shows the trend described in Theorem 3.1, i.e., the total uncertainty of MAC fusion is larger than the OPT fusion but smaller than the CI fusion. In Fig. 2(d), we can see also that NF is too optimistic and therefore, unreliable. Furthermore, MAC is preforming better than ICI.

V. Concluding Remarks

This paper considered the problem of track-to-track fusion under unknown correlations. We proposed a novel fusion method to deliver estimates that are (a) less conservative than the ones obtained by the well-known CI method; (b) as expected from a proper fusion algorithm, conservative than the optimal estimate under known correlations. Our fusion method uses a more complex optimization problem than the CI method to obtain its fused track. However, the impact of extra computational effort of our algorithm should be assessed in the context of performance recovery while maintaining the same communication cost of CI. In this paper, due to the space limitation, we only discussed track-to-track fusion between two sensor stations. For multiple tracks, a sequential fusion method similar to the one proposed for CI fusion in [37] can be used.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] J. K. Uhlmann, "Covariance consistency methods for fault-tolerant distributed data fusion," *Information Fusion*, vol. 4, no. 3, pp. 201–215, Sep. 2003.

[3] C. Y. Chong, S. Mori, K. C. Chang, and W. H. Barker, "Architectures and algorithms for track association and fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 15, no. 1, pp. 5–13, Jan. 2000.

[4] E. Waltz and J. Llinas, *Multisensor Data Fusion*. Norwood, MA: Artech House, 1990.

[5] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, Apr. 1981.

[6] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, no. 6, pp. 803–805, Nov. 1986.

[7] R. A. Singer and A. J. Kanyuck, "Computer control of multiple site track correlation," *Automatica*, vol. 7, no. 4, pp. 455–463, Jul. 1971.

[8] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. New York, NY: Academic Press, 1982.

[9] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, Jan. 2013.

[10] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *IEEE Int. Conf. on Decision and Control*, Shanghai, China, Dec. 2009, pp. 7036–7042.

[11] ——, "Distributed kalman filtering for sensor networks," in *IEEE Int. Conf. on Decision and Control*, New Orleans, LA, USA, Dec. 2007, pp. 5492–5498.

[12] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, Dec. 2013.

[13] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *American Control Conference*, vol. 4, Albuquerque, NM, USA, Jun. 1997, pp. 2369–2373.

[14] ——, *Handbook of Data Fusion*. Boca Raton, FL: CRC Press, 2001, ch. 12, pp. 1–25.

[15] ——, "Using covariance intersection for SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3–20, Jan. 2007.

[16] W. Qi, P. Zhang, and Z. Deng, "Robust sequential covariance intersection fusion Kalman filtering over multi-agent sensor networks with measurement delays and uncertain noise variances," *Acta Automatica Sinica*, vol. 40, no. 11, pp. 2632–2642, Nov. 2014.

[17] G. Wang, N. Li, and Y. Zhang, "Diffusion distributed Kalman filter over sensor networks without exchanging raw measurements," *Signal Processing*, vol. 132, pp. 1–7, Mar. 2017.

[18] B. Chen, G. Hu, D. W. C. Ho, and L. Yu, "Distributed covariance intersection fusion estimation for cyber-physical systems with communication constraints," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4020–4026, Dec. 2016.

[19] J. Hu, L. Xie, and C. Zhang, "Diffusion kalman filtering based on covariance intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891–902, Feb. 2012.

[20] J. Cong, Y. Li, G. Qi, and A. Sheng, "An order insensitive sequential fast covariance intersection fusion algorithm," *Information Sciences*, vol. 367, pp. 28–40, Nov. 2016.

[21] T. Sun, M. Xin, and B. Jia, "Distributed estimation in general directed sensor networks based on batch covariance intersection," in *American Control Conference*, Boston, MA, USA, Jul. 2016, pp. 5492–5497.

[22] A. R. Benaskeur, "Consistent fusion of correlated data sources," in *Annual Conference of the IEEE Industrial Electronics Society*, vol. 4, Sevilla, Spain, Nov. 2002, pp. 2652–2656.

[23] J. Sijs and M. Lazar, "State fusion with unknown correlation: Ellipsoidal intersection," *Automatica*, vol. 48, no. 8, pp. 1874–1878, Aug. 2012.

[24] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Decentralized data fusion with inverse covariance intersection," *Automatica*, vol. 79, pp. 35–41, May 2017.

[25] R. Horst and N. V. Thoai, "DC programming: overview," *Journal of Optimization Theory & Applications*, vol. 103, no. 1, pp. 1–43, Oct. 1999.

[26] T. P. Dinh and H. A. L. Thi, "Recent advances in DC programming and DCA," *Transactions on Computational Collective Intelligence*, vol. 8342, pp. 1–37, 2014.

[27] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[28] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge: Cambridge University Press, 1991.

[29] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: Yaakov Bar-Shalom, 2011.

[30] J. K. Uhlmann, "General data fusion for estimates with unknown cross covariances," in *Aerospace/Defense Sensing and Controls*, vol. 2755, Orlando, FL, USA, Jun. 1996, pp. 536–547.

[31] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge: Cambridge University Press, 2013.

[32] H. Durrant-Whyte, "Introduction to estimation and the Kalman filter," *University of Sydney, Australian Centre for Field Robotics*, vol. 28, no. 3, Jan. 2001.

[33] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY: John Wiley & Sons, 2001.

[34] R. C. Thompson and L. J. Freede, "Eigenvalues of partitioned hermitian matrices," *Bulletin of the Australian Mathematical Society*, vol. 3, no. 1, pp. 23–37, 1970.

[35] X. R. Li, Z. Zhao, and V. P. Jilkov, "Practical measures and test for credibility of an estimator," in *Workshop on Estimation, Tracking, and Fusion-A Tribute to Yaakov Bar-Shalom*, Monterey, CA, USA, May 2001, pp. 481–495.

[36] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *IEEE Int. Conf. on Decision and Control*, Seville, Spain, Dec. 2005, pp. 8179–8184.

[37] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao, "Sequential covariance intersection fusion Kalman filter," *Information Sciences*, vol. 189, pp. 293–309, Apr. 2012.