A Track-to-Track Fusion Method via Construction of Cross-Covariance Matrix for Tracks with Unknown Correlations

Mahboubeh Zarei-Jalalabadi[a,b,*], Solmaz S. Kia[b], Seyed Mohammad-Bagher Malaek[a]

[a] *Aerospace Engineering Department, Sharif University of Technology, Tehran, Iran.*

[b] *Mechanical and Aerospace Engineering Department, University of California Irvine, Irvine, CA 2697.*

Abstract

This paper considers 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. In particular, we take advantage of the properties of positive definite matrices to show that the cross-covariance matrix of two tracks to be fused is given by a structured matrix consisted of product of square roots of the tracks' covariance matrices and a contraction matrix. Then, we propose an appropriate optimization problem to obtain this contraction matrix in a way that the fused track is less conservative than the one obtained by covariance intersection 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 and also show how its design optimization problem can be cast as a difference of convex (DC) programing problem, which can be solved in an efficient manner using DC programing software solutions. We evaluate the performance of the proposed method through analysing both root mean squared error and average normalised estimation error squared tests over Monte Carlo runs for a two-dimensional system.

Keywords: Track-to-Track Fusion, Kalman Filtering, Data Correlation, Covariance Intersection, Difference of Convex Programming

Email addresses: `zareijalalabadi_m@ae.sharif.edu` (Mahboubeh Zarei-Jalalabadi), `solmaz@uci.edu` (Solmaz S. Kia), `malaek@sharif.edu` (Seyed Mohammad-Bagher Malaek)

*Corresponding author. This research was conducted during the time the first author was a visiting Ph.D. student in the University of California Irvine.

Email addresses: `zareijalalabadi_m@ae.sharif.edu` (Mahboubeh Zarei-Jalalabadi), `solmaz@uci.edu` (Solmaz S. Kia), `malaek@sharif.edu` (Seyed Mohammad-Bagher Malaek)

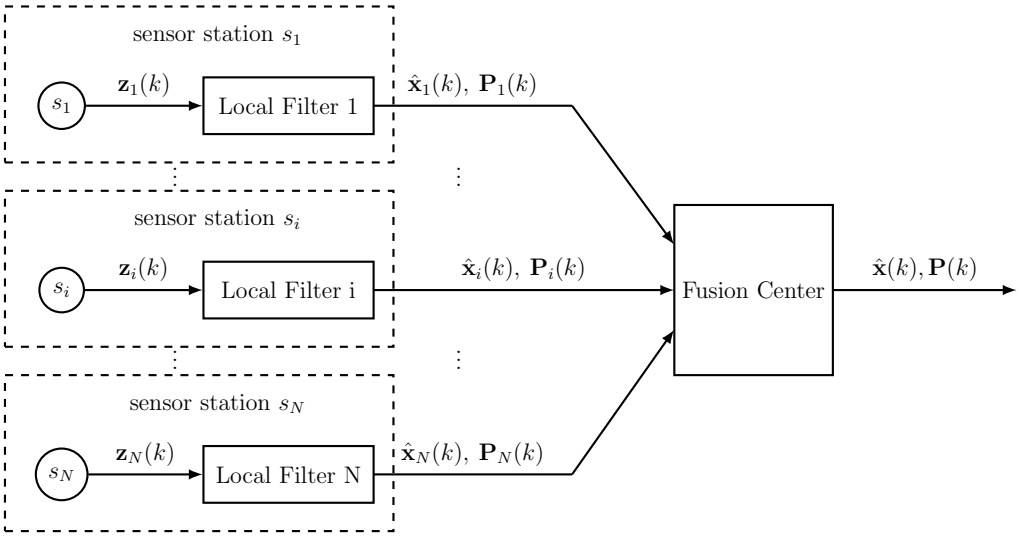


Figure 1 – A functional architecture for track-to-track fusion in which sensor stations transmit local estimates to a fusion center where the fused estimate is assimilated.

1. Introduction

Track-to-track fusion refers to the process of combining multiple noise-corrupted estimates together to obtain an improved estimate of the underlying states of an existing estimation architecture. Unlike the data fusion algorithms that combine observations from different sensors, in track-to-track fusion algorithms, local sensor stations generate local estimates using local filters. The local estimates are then transmitted to a central fusion site and are combined together to deliver the so called "fused estimate" . Figure 1 illustrates a typical track-to-track fusion architecture. Track-to-track fusion, once executed in an appropriate manner, yields more accurate estimates than using a single sensor [1, 2, 3].

Although track-to-track fusion algorithms can deliver a better estimate, they are faced with a major challenge on how to keep an accurate account of the correlation between the local estimates of the sensors without resistant communication with sensor stations at each time step. In general, the local estimates are correlated due to a common process noise that enters into the estimation error of the local estimates [4]. In earlier work on track-to-track fusion it was assumed that there was no correlation between two local estimates [5]. Such an assumption however leads to fused estimates that are too optimistic. This property is referred to as the inconsistency phenomenon which in extreme cases can even lead to filter divergence as reported in [6],[7]. To overcome the problem of inconsistency in the track-to-track fusion, the correlations between the tracks should be accounted for in either explicit or implicit manner. A short review of some of the existing well-established techniques is provided in the proceeding section.

Keeping track of exact correlation between sensor stations requires persistent transmission of the local tracks by the sensor stations to the fusion center. Such a requirement may not

be realizable due to communication cost and also communication failures. Therefore, there has been a great interest in track-to-track fusion methods that can account for correlation terms in an implicit manner. The prime example of such techniques is the 'covariance intersection' method, initially proposed in [8]. In this method the lack of knowledge of correlation between the tracks is compensated by ensuring that the resulting estimate is conservative. Covariance intersection method has been widely used in decentralised sensor network problems [9, 10, 11, 12]. However, this method often results in highly conservative estimates, i.e., the estimated covariance can be much larger than the actual covariance.

In this paper, we propose a novel method to construct the correlation terms between tracks from two sensors. Our objective is to trade in extra computation for better fusion performance. We start by observing that the cross-covariance matrix of joint correlated covariance matrix of two tracks can be stated as the product of the square root of the covariance matrices of the tracks and a contraction matrix. We 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 covariance intersection method. Moreover, we show that our fused track, as expected from an approximation technique, is more conservative that the optimal track one can obtain by taking into account the exact value of the cross-covariance between the tracks. Through rigorous analysis, we prove our new fusion algorithm's properties and also show how its design optimization problem can be cast as a difference of convex (DC) programing problem (cf. [13]), which can be solved in an efficient manner using DC programing software solutions. We also show parallels between the structure we use to estimate the cross-covariance matrix and the estimated cross-covariance one can calculate using sigma points generated via Unscented Kalman filter (UKF) type approach (see [14, 15]).

The remainder of this paper is organised as follows: In Section 2, we introduce our notations and review some preliminary results and definitions which we use through out the paper. In Section 3, we introduce our objective statements. Section 4 presents the details of our proposed track-to-track fusion method. In section 5, we evaluate the performance of our proposed algorithm in simulations. Finally, section 6 presents our conclusions.

2. Preliminaries

In this section, we introduce basic notations and review some relevant materials from literature that we use to define and formulate our problem.

We let $\mathbb{R}$ to denote the set of real numbers. The set of $n \times n$ real positive semi definite and positive definite matrices are, respectively, $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$. The $n \times m$ zero matrix is $\mathbf{0}_{n\times m}$ (when $m = n$, we use $\mathbf{0}_n$) while the $n \times n$ identity matrix is $\mathbf{I}_n$. The transpose of matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ is $\mathbf{A}^\top$. For a matrix $\mathbf{A} \in \mathbb{S}^n_+$ its matrix square root is $\sqrt{\mathbf{A}}$ which satisfies $\sqrt{\mathbf{A}}^\top \sqrt{\mathbf{A}} = \mathbf{A}$. We denote the Kronecker delta by $\delta(k, l)$, that is, $\delta(k, l) = 0$, if $l \neq k$ and $\delta(k, l) = 1$, if $l = k$.

In the following, for the convenience of the reader, we review some of the mathematical definitions and relations that we invoke to develop our proposed track-to-track fusion algorithm.

Definition 2.1 (Contraction matrix)**.** A matrix $\mathbf{C} \in \mathbb{R}^{n\times m}$ is called a contraction if its spectral norm satisfies $\|\mathbf{C}\|_2 \leq 1$; it is a strict contraction if $\|\mathbf{C}\|_2 < 1$. We represent the set of real contraction matrices with dimension $n \times m$ by $\mathcal{C}^{n\times m}$. □

The next result uses the properties of the contraction matrices to introduce a relationship among the diagonal and off diagonal elements of a 2×2 block partitioned matrix to guarantee that the joint matrix is semi-positive definite or positive definite matrix.

Lemma 2.1 (c.f. [16, 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

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{B} \end{bmatrix} \tag{1}$$

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 $\mathbf{C}\in\mathcal{C}^{m\times n}$ such that

$$\mathbf{X} = \sqrt{\mathbf{A}}^\top \mathbf{C} \sqrt{\mathbf{B}}. \tag{2}$$

□

In our developments below we use DC programing to solve an optimization criterion of our algorithm. We close this section by reviewing the definition of difference of convex functions.

Definition 2.2 (Difference of convex function [17])**.** The real valued function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be difference of convex (DC) function if 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

$$f(\mathbf{x}) = g(\mathbf{x}) - h(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

□

3. Objective statement

In this section, we formulate our problem of interest. We start by a short review of the track-to-track fusion problem to introduce notations and equations needed later in our proposed algorithm. Then, to provide a context for our objective, we review two of the existing methods for track-to-track fusion. We close this section by stating our objective.

Consider sensor stations s_i and s_j, each with processing power, observing and estimating a time-varying target/process whose true state is denoted by $\mathbf{x} \in \mathbb{R}^n$. Under the Gaussian noise assumption for measurement and filtering, at each time k sensor stations s_i and s_j deliver their respective estimate and its corresponding error covariance ($\hat{\mathbf{x}}_r(k) \in \mathbb{R}^n$, $\mathbf{P}_r(k) \in \mathbb{S}^n_{++}$), $r = \{i, j\}$, to a fusion center. Through out the paper, we assume that the tracks that are being fused at the fusion center are synchronised. To simplify the notation, in the

following, we drop the time index k of these estimates. Because tracks from sensors s_i and s_j are originated from same process whose process noise is common for both sensors, these tracks are correlated, i.e., the cross-covariance $\mathbf{P}_{ij}$ between estimates of these two sensors is non-zero [18].

The fusion center uses the linear combination of the estimates of sensors s_i and s_j to obtain an improved estimate ($\hat{\mathbf{x}}_c \in \mathbb{R}^n$, $\mathbf{P}_c \in \mathbb{S}^n_{++}$), i.e.,

$$\hat{\mathbf{x}}_c = \mathbf{W}_i \hat{\mathbf{x}}_i + \mathbf{W}_j \hat{\mathbf{x}}_j, \tag{3}$$

where in order to have a zero mean error, $\mathbf{E}[\mathbf{x} - \hat{\mathbf{x}}_c] = \mathbf{0}_{n\times 1}$, the weights are chosen such that

$$\mathbf{W}_i + \mathbf{W}_j = \mathbf{I}_n. \tag{4}$$

Then, the gains are computed using the following optimization to arrive at a consistent minimum variance estimate

$$\mathbf{W} = \operatorname{argmin} \det\big(\mathbf{P}_c\big) \quad \text{subject to} \quad \mathbf{W} \begin{bmatrix} \mathbf{I}_n \\ \mathbf{I}_n \end{bmatrix} = \mathbf{I}_n, \tag{5}$$

where $\mathbf{W} \triangleq \begin{bmatrix} \mathbf{W}_i & \mathbf{W}_j \end{bmatrix}$, and

$$\begin{aligned} \mathbf{P}_c &= \mathbf{E}\big[(\mathbf{x} - \hat{\mathbf{x}}_c)\,(\mathbf{x} - \hat{\mathbf{x}}_c)^\top\big] \\ &= \mathbf{W}_i \mathbf{P}_i \mathbf{W}_i^\top + \mathbf{W}_j \mathbf{P}_j \mathbf{W}_j^\top + \mathbf{W}_i \mathbf{P}_{ij} \mathbf{W}_j^\top + \mathbf{W}_j \mathbf{P}_{ij}^\top \mathbf{W}_i^\top = \mathbf{W}\,\mathbf{P}_J\,\mathbf{W}^\top, \end{aligned} \tag{6}$$

where $\mathbf{P}_J$ is the joint covariance matrix

$$\mathbf{P}_J \triangleq \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_{ij} \\ \mathbf{P}_{ij}^\top & \mathbf{P}_j \end{bmatrix}. \tag{7}$$

When the cross-covariance $\mathbf{P}_{ij}$ between the estimate of sensor s_i and that of sensor s_j is known, the consistent minimum variance fused estimate is given by the following result.

Theorem 3.1 (c.f. [18], [8])**.** Given $\mathbf{P}_i$, $\mathbf{P}_j$, and $\mathbf{P}_{ij}$, the optimal solution of the weighting matrix $\mathbf{W}$ in the constrained optimization problem (5) yields a consistent estimate $\hat{\mathbf{x}}_c^\star$ with covariance $\mathbf{P}_c^\star$ as follows

$$\hat{\mathbf{x}}_c^\star = \hat{\mathbf{x}}_i + \Big(\mathbf{P}_i - \mathbf{P}_{ij}\Big)\Big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{P}_{ij} - \mathbf{P}_{ij}^\top\Big)^{-1}\Big(\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_i\Big), \tag{8a}$$

$$\mathbf{P}_c^\star = \mathbf{P}_i - \Big(\mathbf{P}_i - \mathbf{P}_{ij}\Big)\Big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{P}_{ij} - \mathbf{P}_{ij}^\top\Big)^{-1}\Big(\mathbf{P}_i - \mathbf{P}_{ij}^\top\Big). \tag{8b}$$

□

Cross-covariance propagation method for track-to-track fusion: when the cross-covariance matrix is known at each time step k, then the fusion equations (8) yield the consistent

minimum variance fused estimate of the process. For linear processes whose state equations are described by

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \boldsymbol{\omega}(k), \qquad \mathbf{E}[\boldsymbol{\omega}(k)\boldsymbol{\omega}^\top(l)] = \mathbf{Q}(k)\,\delta(k,l), \tag{9a}$$

$$\mathbf{z}_r(k) = \mathbf{H}_r(k)\mathbf{x}(k) + \boldsymbol{\nu}_r(k), \quad \mathbf{E}[\boldsymbol{\nu}_r(k)\boldsymbol{\nu}_r^\top(l)] = \mathbf{R}_r(k)\,\delta(k,l), \quad r = \{i,j\}, \tag{9b}$$

where $\mathbf{E}[\boldsymbol{\nu}_i(k)\boldsymbol{\nu}_j^\top(k)] = \mathbf{0}_n$, [19] has shown that the exact cross-covariance matrix $\mathbf{P}_{ij}(k)$ is propagated through time using the following recursive formula

$$\mathbf{P}_{ij}(k) = \Big(\mathbf{I}_n - \mathbf{K}_i(k)\mathbf{H}_i(k)\Big)\Big(\mathbf{F}(k-1)\mathbf{P}_{ij}(k-1)\mathbf{F}^\top(k-1) + \mathbf{Q}(k-1)\Big)\Big(\mathbf{I}_n - \mathbf{K}_j(k)\mathbf{H}_j(k)\Big)^\top. \tag{10}$$

Here, $\mathbf{K}_i(k)$ and $\mathbf{K}_j(k)$ are Kalman gains at sensor stations s_i and s_j, respectively. Under the assumption that the initial measurements are uncorrelated, the initial condition of (10) is set to zero (c.f. [19] for further details). □

Although equation (10) computes the exact cross-covariance between the estimates of the two sensor stations s_i and s_j, this method is restricted to applications that sensors implement the Kalman filter. Moreover, because (10) is a recursive equation, the cross-covariance propagation method requires the transmission of the sensor tracks 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.

Covariance intersection method for track-to-track fusion: covariance intersection method, proposed in [8] (see also [20]), is a track-to-track fusion method that does not require an explicit knowledge of the cross-covariance of the sensors' tracks. For tracks $(\hat{\mathbf{x}}_i\ , \mathbf{P}_i)$ and $(\hat{\mathbf{x}}_j\ , \mathbf{P}_j)$ with unknown correlation, covariance intersection method fuses the tracks according to

$$\hat{\mathbf{x}}_{\mathrm{CI}} = \mathbf{P}_{\mathrm{CI}}\Big(\omega\mathbf{P}_i^{-1}\hat{\mathbf{x}}_i + (1-\omega)\mathbf{P}_j^{-1}\hat{\mathbf{x}}_j\Big), \tag{11a}$$

$$\mathbf{P}_{\mathrm{CI}}^{-1} = \omega\,\mathbf{P}_i^{-1} + (1-\omega)\,\mathbf{P}_j^{-1}, \tag{11b}$$

where $0 \le \omega \le 1$ is a weighting factor that can be used to optimise the updates with respect to a performance criterion, such as minimizing the trace or the determinant of $\mathbf{P}_{\mathrm{CI}}$, e.g.,

$$\omega = \operatorname{argmin} \det\big(\mathbf{P}_{\mathrm{CI}}\big), \quad \text{subject to} \quad 0 \le \omega \le 1. \tag{12}$$

Given (11b), one can see that the optimization algorithm (12) is guaranteed to produce a fused estimate that satisfies $\det(\mathbf{P}_{\mathrm{CI}}) \le \min\big(\det(\mathbf{P}_i), \det(\mathbf{P}_j)\big)$. Moreover, for consistent estimates $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$, the fused estimate $\hat{\mathbf{x}}_{\mathrm{CI}}$ is guaranteed to be consistent [21]. However, the covariance intersection method is a suboptimal fusion method compared to the method (8), indeed, $\mathbf{P}_{\mathrm{CI}} \ge \mathbf{P}_c$ for all $\mathbf{P}_c$ satisfying (6). More precisely, as shown in [19], the error ellipsoid[1]

[1]For a state $\mathbf{x} \in \mathbb{R}^n$ with estimate $\hat{\mathbf{x}} \in \mathbb{R}^n$ and corresponding error covariance $\mathbf{P} \in \mathbb{S}^n_{++}$, the $\kappa\,\sigma$-error ellipsoid is $\{\mathbf{y} \in \mathbb{R}^n \,|\, (\mathbf{y}-\hat{\mathbf{x}})^\top\mathbf{P}^{-1}(\mathbf{y}-\hat{\mathbf{x}}) = \kappa^2\}$, where κ is a constant chosen for a particular confidence threshold, and $\mathbf{y}$ is a point on the ellipsoid boundary [22].

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 Figure 2 for some numerical examples). An interesting relationship between the covariance intersection method and the optimal fusion algorithm (8) is discussed in [21]. There, it is shown that because

$$\begin{bmatrix} \frac{1}{\omega}\mathbf{P}_i & \mathbf{0}_n \\ \mathbf{0}_n & \frac{1}{1-\omega}\mathbf{P}_j \end{bmatrix} \geq \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_{ij} \\ \mathbf{P}_{ij}^\top & \mathbf{P}_j \end{bmatrix} = \mathbf{P}_J \ , \tag{13}$$

holds for any $0 \leq \omega \leq 1$, one can assume that the covariance intersection method is a solution of (8), in which, the joint covariance matrix $\mathbf{P}_J$ of the fused tracks has been replaced by the consistent but conservative overestimated joint error covariance matrix in the left hand side of (13). □

Covariance intersection method due to its low communication cost and its robustness to communication failure has a great appeal for track-to-track fusion over sensor networks. However, as discussed above, this suboptimal fusion strategy can be very conservative. Our objective in this paper is to propose an alternative track-to-track fusion algorithm which similar to the covariance intersection method does not require the exact knowledge of cross-covariances of the tracks, however, it produces a fused track that is less conservative. For reasons that will be clear in the next section, we refer to our algorithm as Maximum Allocated Covariance (MAC). We state our objective as follow.

Objective: For two unbiased and consistent tracks $(\hat{\mathbf{x}}_i \in \mathbb{R}^n, \mathbf{P}_i \in \mathbb{S}^n_{++})$ and $(\hat{\mathbf{x}}_j \in \mathbb{R}^n, \mathbf{P}_j \in \mathbb{S}^n_{++})$ whose correlation is unknown, obtain the fused track $(\hat{\mathbf{x}}_{\text{MAC}} \in \mathbb{R}^n, \mathbf{P}_{\text{MAC}} \in \mathbb{S}^n_{++})$ such that $\hat{\mathbf{x}}_{\text{MAC}}$ is a weighted linear fused track

$$\hat{\mathbf{x}}_{\text{MAC}} = \mathbf{W}_i\hat{\mathbf{x}}_i + \mathbf{W}_j\hat{\mathbf{x}}_j, \tag{14}$$

which satisfies

$$\mathbf{E}[\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}}] = \mathbf{0}_{n\times 1}, \tag{15a}$$

$$\mathbf{0}_n \leq \mathbf{P}_{\text{MAC}} \leq \mathbf{P}_i \ , \text{ and } \mathbf{0}_n \leq \mathbf{P}_{\text{MAC}} \leq \mathbf{P}_j \ , \tag{15b}$$

$$\det(\mathbf{P}_c^\star) \leq \det(\mathbf{P}_{\text{MAC}}) \leq \det(\mathbf{P}_{\text{CI}}^\star), \tag{15c}$$

where $\mathbf{P}_c^\star$ is given in (8) and $\mathbf{P}_{\text{CI}}^\star$ satisfies (11b) with optimum weights determined from (12). □

Here, condition (15a) requires $\hat{\mathbf{x}}_{\text{MAC}}$ to be unbiased, while (15b) seeks a fusion policy which produces better estimate than individual tracks of the sensor stations s_i and s_j. The lower bound on the condition (15c) ensures that the fused track is not optimistic in comparison to the optimal fused track, while the upper bound requires the fused track to be less conservative than the track obtained from the covariance intersection fusion. Here, we use $\det(\mathbf{P})$ as the scalar measure to compare the total uncertainty of our proposed method to others.

4. Maximum Allocated Covariance

In this section, we propose a novel track-to-track fusion algorithm for tracks with unknown correlation. The design of this method 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. Subsequently, the ellipsoid corresponding to the fused covariance matrix will exactly fit within the intersection region defined by the ellipsoids corresponding to the two local covariance matrices. Therefore, MAC improves the performance of the estimation and avoids overestimation, simultaneously. To obtain such fusion rule, the proposed method performs two tasks; 1) it exploits an appropriate structure to construct the cross-covariance matrix that contains an unknown tuning matrix and 2) finds the unknown tuning matrix through a maximization problem such that the fused estimate is consistent in the sense that the determinant of the fused covariance is greater than the optimal value provided by the minimum variance filter (8) and smaller than the result CI provides.

We start by providing some preliminary results that are used in developing our algorithm. For two given tracks $(\hat{\mathbf{x}}_i \in \mathbb{R}^n, \mathbf{P}_i \in \mathbb{S}^n_{++})$ and $(\hat{\mathbf{x}}_j \in \mathbb{R}^n, \mathbf{P}_j \in \mathbb{S}^n_{++})$ corresponding respectively to the sensor stations s_i and s_j, we define the set of all matrices $\mathbf{X} \in \mathbb{R}^{n\times n}$ that make the joint matrix $\begin{bmatrix} \mathbf{P}_i & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{P}_j \end{bmatrix}$ positive define by $\mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j)$, i.e.,

$$\mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j) = \left\{ \mathbf{X} \in \mathbb{R}^{n\times n} \middle| \begin{bmatrix} \mathbf{P}_i & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{P}_j \end{bmatrix} \in \mathbb{S}^{2n}_{++} \right\}. \tag{16}$$

The next results invoke Lemma 2.1 to relate the unknown correlations of the tracks from sensor s_i and s_j with their known covariances $\mathbf{P}_i$ and $\mathbf{P}_j$.

Lemma 4.1. For two given tracks $(\hat{\mathbf{x}}_i \in \mathbb{R}^n, \mathbf{P}_i \in \mathbb{S}^n_{++})$ and $(\hat{\mathbf{x}}_j \in \mathbb{R}^n, \mathbf{P}_j \in \mathbb{S}^n_{++})$, the following holds

$$\mathbf{X} \in \mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j) \Leftrightarrow \exists\, \mathbf{C} \in \mathcal{C}^{n\times n},\ \text{such that } \mathbf{X} = \sqrt{\mathbf{P}_i}^\top \mathbf{C} \sqrt{\mathbf{P}_j}\,. \tag{17}$$

Proof. The proof is through the straightforward application of Lemma 2.1. □

Corollary 4.1. For two correlated given tracks $(\hat{\mathbf{x}}_i \in \mathbb{R}^n, \mathbf{P}_i \in \mathbb{S}^n_{++})$ and $(\hat{\mathbf{x}}_j \in \mathbb{R}^n, \mathbf{P}_j \in \mathbb{S}^n_{++})$, thier cross-covariance matrix $\mathbf{P}_{ij}$ satisfies

$$\mathbf{P}_{ij} \in \mathcal{J}_{++}(\mathbf{P}_i\ , \mathbf{P}_j) \text{ and } \exists\, \mathbf{C} \in \mathcal{C}^{n\times n},\ \text{such that } \mathbf{P}_{ij} = \sqrt{\mathbf{P}_i}^\top \mathbf{C} \sqrt{\mathbf{P}_j}\,. \tag{18}$$

Proof. Recall that by definition the joint covariance matrix of tracks of sensors $\mathbf{P}_i$ and $\mathbf{P}_j$ is positive definite. Then, the proof becomes an immediate result of Lemma 4.1. □

We are now ready to present our main result which states our fusion rule that satisfies our objective stated in Section 3.

Theorem 4.1 (MAC track-to-track fusion)**.** Consider two unbiased and consistent tracks $(\hat{\mathbf{x}}_i \in \mathbb{R}^n, \mathbf{P}_i \in \mathbb{S}^n_{++})$ and $(\hat{\mathbf{x}}_j \in \mathbb{R}^n, \mathbf{P}_j \in \mathbb{S}^n_{++})$, with an unknown correlation from, respectively, sensor stations s_i and s_j. Let $\mathbf{X}^\star = \sqrt{\mathbf{P}_i}^\top \mathbf{C}^\star \sqrt{\mathbf{P}_j}$ with

$$\mathbf{C}^* = \underset{\mathbf{C} \in \mathcal{C}^{n\times n}}{\operatorname{argmax}} \det\left(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top \mathbf{C} \sqrt{\mathbf{P}_j})\right), \tag{19}$$

where

$$\mathbf{F}(\mathbf{X}) = \mathbf{P}_i - (\mathbf{P}_i - \mathbf{X})(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X} - \mathbf{X}^\top)^{-1}(\mathbf{P}_i - \mathbf{X}^\top). \tag{20}$$

Then, the fused track $(\hat{\mathbf{x}}_{\text{MAC}}, \mathbf{P}_{\text{MAC}})$

$$\hat{\mathbf{x}}_{\text{MAC}} = \mathbf{W}_i \hat{\mathbf{x}}_i + \mathbf{W}_j \hat{\mathbf{x}}_j, \tag{21a}$$

$$\mathbf{P}_{\text{MAC}} = \mathbf{F}(\mathbf{X}^\star) = \mathbf{P}_i - \left(\mathbf{P}_i - \mathbf{X}^\star\right)\left(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\right)^{-1}\left(\mathbf{P}_i - \mathbf{X}^{\star\top}\right), \tag{21b}$$

where

$$\mathbf{W}_i = \mathbf{I}_n - \mathbf{W}_j = \left(\mathbf{P}_j - \mathbf{X}^{\star\top}\right)\left(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\right)^{-1}, \tag{22a}$$

$$\mathbf{W}_j = \left(\mathbf{P}_i - \mathbf{X}^\star\right)\left(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\right)^{-1}, \tag{22b}$$

satisfies our objectives stated in (15).

Proof. Because the tracks from sensor stations s_i and s_j are unbiased and also the gains in (22) satisfy $\mathbf{W}_i + \mathbf{W}_j = \mathbf{I}_n$, objective (15a) is satisfied as shown below

$$\begin{aligned}
\mathbf{E}[\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}}] &= \mathbf{E}[(\mathbf{W}_i + \mathbf{W}_j)\mathbf{x} - (\mathbf{W}_i\hat{\mathbf{x}}_i + \mathbf{W}_j\hat{\mathbf{x}}_j)] \\
&= \mathbf{E}[\mathbf{W}_i(\mathbf{x} - \hat{\mathbf{x}}_i)] + \mathbf{E}[\mathbf{W}_j(\mathbf{x} - \hat{\mathbf{x}}_j)] \\
&= \mathbf{0}_{n\times 1} + \mathbf{0}_{n\times 1} = \mathbf{0}_{n\times 1}.
\end{aligned}$$

Next, we show the validity of statement (15b). To this end, notice that $\mathbf{X}^\star$ by its construction and due to Lemma 4.1, satisfies $\mathbf{X}^\star \in \mathcal{J}_{++}$. As a result the joint constructed covariance matrix is guaranteed to be positive definite, i.e.,

$$\begin{bmatrix} \mathbf{P}_i & \mathbf{X}^\star \\ \mathbf{X}^{\star\top} & \mathbf{P}_j \end{bmatrix} \in \mathbb{S}^{2n}_{++}. \tag{23}$$

Next, we apply a congruent transformation with non-singular transformation matrix $\begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n \\ \mathbf{0}_n & -\mathbf{I}_n \end{bmatrix}$ to the constructed joint covariance matrix to obtain

$$\begin{bmatrix} \mathbf{P}_i & \mathbf{X}^\star \\ \mathbf{X}^{\star\top} & \mathbf{P}_j \end{bmatrix} \geq \mathbf{0}_n \Rightarrow \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_n \\ \mathbf{I}_n & -\mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{P}_i & \mathbf{X}^\star \\ \mathbf{X}^{\star\top} & \mathbf{P}_j \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n \\ \mathbf{0}_n & -\mathbf{I}_n \end{bmatrix} = \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_i - \mathbf{X}^\star \\ \mathbf{P}_i - \mathbf{X}^{\star\top} & \mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top} \end{bmatrix} \geq \mathbf{0}_n,$$

which by virtue of the properties of Schur complement of a positive definite matrices gives us the guarantees that

$$\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top} \geq \mathbf{0}_n, \tag{24a}$$

$$\mathbf{P}_i - \Big(\mathbf{P}_i - \mathbf{X}^\star\Big)\Big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\Big)^{-1}\Big(\mathbf{P}_i - \mathbf{X}^{\star\top}\Big) \geq \mathbf{0}_n, \tag{24b}$$

From (24) we conclude that $\mathbf{P}_{\text{MAC}} \in \mathbb{S}^n_{++}$ and also that

$$\Big(\mathbf{P}_i - \mathbf{X}^\star\Big)\Big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\Big)^{-1}\Big(\mathbf{P}_i - \mathbf{X}^{\star\top}\Big) \geq \mathbf{0}_n,$$

which gives us

$$\mathbf{P}_i - \mathbf{P}_{\text{MAC}} = \Big(\mathbf{P}_i - \mathbf{X}^\star\Big)\Big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\Big)^{-1}\Big(\mathbf{P}_i - \mathbf{X}^{\star\top}\Big) \geq \mathbf{0}_n.$$

To complete our proof of objective (15b), remark that we can write $\mathbf{P}_{\text{MAC}}$ as

$$\mathbf{P}_{\text{MAC}} = \mathbf{P}_j - \Big(\mathbf{P}_j - \mathbf{X}^{\star\top}\Big)\Big(\mathbf{P}_i + \mathbf{P}_j - \mathbf{X}^\star - \mathbf{X}^{\star\top}\Big)^{-1}\Big(\mathbf{P}_j - \mathbf{X}^\star\Big).$$

Therefore, following the similar steps as above but with congruent transformation matrix $\begin{bmatrix} -\mathbf{I}_n & \mathbf{0}_n \\ \mathbf{I}_n & \mathbf{I}_n \end{bmatrix}$, 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 that our objective (15c) is satisfied by $(\hat{\mathbf{x}}_{\text{MAC}}, \mathbf{P}_{\text{MAC}})$ given in the statement, i.e., $\det(\mathbf{P}^\star_c) \leq \det(\mathbf{P}_{\text{MAC}}) \leq \det(\mathbf{P}^\star_{\text{CI}})$. To prove the lower bound, notice that since $\mathbf{P}_{ij} \in \mathcal{J}_{++}$ (recall Corollary 4.1), then

$$\mathbf{P}^\star_c = \mathbf{F}(\mathbf{P}_{ij}),$$

where $\mathbf{F}$ is given in (20). Since in optimization problem (19), we are maximizing $\det \mathbf{F}(\mathbf{X})$ over all possible $\mathbf{X} \in \mathcal{J}_{++}$ to obtain $\mathbf{P}_{\text{MAC}} = \mathbf{F}(\mathbf{X}^\star)$, we have then $\det(\mathbf{P}^\star_c) \leq \det(\mathbf{P}_{\text{MAC}})$. To show that $\det(\mathbf{P}_{\text{MAC}}) \leq \det(\mathbf{P}^\star_{\text{CI}})$ we show that $\mathbf{P}_{\text{CI}} \geq \mathbf{P}_{\text{MAC}}$ as follows. We have already shown that $\mathbf{P}_{\text{MAC}} \leq \mathbf{P}_i$ which implies $\mathbf{P}^{-1}_{\text{MAC}} \geq \mathbf{P}^{-1}_i$. From (11b), we also have $\mathbf{P}^{-1}_{\text{CI}} \geq \omega \mathbf{P}^{-1}_i$, where $0 \leq \omega \leq 1$. Therefore we can conclude the following

$$\mathbf{P}^{-1}_{\text{MAC}} - \mathbf{P}^{-1}_{\text{CI}} \geq (1-\omega)\mathbf{P}^{-1}_i \geq \mathbf{0}_n, \tag{25}$$

which gives us $\mathbf{P}_{\text{MAC}} \leq \mathbf{P}_{\text{CI}}$. This completes our proof. □

For three numerical examples, figure 2 demonstrates the uncertainty ellipsoid of MAC's fused track compared to the one corresponding to the covariance intersection method and that of the optimal fusion (8), which uses the exact value of the cross-covariance between the local sensor tracks. As seen, in these examples, our fusion objectives (15b) and (15c) are satisfied.

Our next result shows that the optimization problem (19) can be cast in an equivalent DC programing optimization form, enabling us to obtain MAC track-to-track fusion in an efficient manner using DC programing software solutions.

Theorem 4.2 (MAC is a DC programing problem)**.** The optimization problem (19) as defined in Theorem 4.1 is equivalent to the DC programing problem below

$$\begin{aligned}\mathbf{C}^* =& \underset{\mathbf{C}\in\mathcal{C}^{n\times n}}{\operatorname{argmax}}\Big(\operatorname{Log}\big(\det(\mathbf{I}_n-\mathbf{C}^\top\mathbf{C})\big)\\ &-\operatorname{Log}\big(\det(\mathbf{P}_i+\mathbf{P}_j-\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}-\sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i})\big)\Big).\end{aligned} \tag{26}$$

Proof. First, we show that optimization problem (26) is a DC programing problem. To this end, notice that the first logarithm expression in (26) is a concave function in matrix $\mathbf{C}$, and the second term considering its sign is a convex function. Therefore based on definition 2.2 the function defined in (33) is a DC function. The constraint can be written as $\mathbf{C}^\top\mathbf{C}-\mathbf{I}_n<\mathbf{0}_n$ which is a convex constraint. Thus, the optimization problem (26) is a DC programing problem. Next, we show that optimization problem (19) is equivalent to (26).

Consider the following matrix

$$\overline{\mathbf{P}}=\begin{bmatrix}\mathbf{P}_i+\mathbf{P}_j-\mathbf{X}-\mathbf{X}^\top & \mathbf{P}_i-\mathbf{X}^\top\\ \mathbf{P}_i-\mathbf{X} & \mathbf{P}_i\end{bmatrix}. \tag{27}$$

The Schur complement of the block $\mathbf{P}_i+\mathbf{P}_j-\mathbf{X}-\mathbf{X}^\top$ of the matrix $\overline{\mathbf{P}}$ is

$$\mathbf{P}_i-\Big(\mathbf{P}_i-\mathbf{X}\Big)\Big(\mathbf{P}_i+\mathbf{P}_j-\mathbf{X}-\mathbf{X}^\top\Big)^{-1}\Big(\mathbf{P}_i-\mathbf{X}^\top\Big)=\mathbf{F}(\mathbf{X}) \tag{28}$$

To obtain $\det(\mathbf{F}(\mathbf{X}))$ of optimization problem (19) we proceed as follows. We use the relation between determinant of a matrix and its Schur complement to write

$$\det(\overline{\mathbf{P}})=\det(\mathbf{F}(\mathbf{X}))\det(\mathbf{P}_i+\mathbf{P}_j-\mathbf{X}-\mathbf{X}^\top) \tag{29}$$

Next, we notice that, by adding and subtracting of the rows and columns of (27), we can show that

$$\det(\overline{\mathbf{P}})=\det(\begin{bmatrix}\mathbf{P}_i & \mathbf{X}\\ \mathbf{X}^\top & \mathbf{P}_j\end{bmatrix}). \tag{30}$$

Then from (28), (29) and (30) and recalling that $\mathbf{X}=\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}$, $\mathbf{P}_r=\sqrt{\mathbf{P}_r}^\top\sqrt{\mathbf{P}_r}$, $r=\{i,j\}$ we can write

$$\det(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}))=\frac{\det(\begin{bmatrix}\sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_i} & \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}\\ \sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i} & \sqrt{\mathbf{P}_j}^\top\sqrt{\mathbf{P}_j}\end{bmatrix})}{\det(\mathbf{P}_i+\mathbf{P}_j-\sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}-\sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i})}. \tag{31}$$

Because we have

$$\begin{bmatrix}\sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_i} & \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j}\\ \sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i} & \sqrt{\mathbf{P}_j}^\top\sqrt{\mathbf{P}_j}\end{bmatrix}=\begin{bmatrix}\sqrt{\mathbf{P}_i}^\top & \mathbf{0}_n\\ \mathbf{0}_n & \sqrt{\mathbf{P}_j}^\top\end{bmatrix}\begin{bmatrix}\mathbf{I}_n & \mathbf{C}\\ \mathbf{C}^\top & \mathbf{I}_n\end{bmatrix}\begin{bmatrix}\sqrt{\mathbf{P}_i} & \mathbf{0}_n\\ \mathbf{0}_n & \sqrt{\mathbf{P}_j}\end{bmatrix}.$$

then we can write

$$\det(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top \mathbf{C}\sqrt{\mathbf{P}_j})) = \frac{\det(\sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_i})\det(\sqrt{\mathbf{P}_j}^\top\sqrt{\mathbf{P}_j})\det(\mathbf{I}_n - \mathbf{C}^\top\mathbf{C})}{\det(\sqrt{\mathbf{P}_i}^\top\sqrt{\mathbf{P}_i} + \sqrt{\mathbf{P}_j}^\top\sqrt{\mathbf{P}_j} - \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j} - \sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i})}. \tag{32}$$

Using chain rule, one can show that the stationary points of problem (32) are equivalent to the stationary points of $\mathrm{Log}(\det(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top \mathbf{C}\sqrt{\mathbf{P}_j})))$.

$$\begin{aligned}\mathrm{Log}(\det(\mathbf{F}(\sqrt{\mathbf{P}_i}^\top \mathbf{C}\sqrt{\mathbf{P}_j}))) =& \mathrm{Log}(\det(\mathbf{P}_i)) + \mathrm{Log}(\det(\mathbf{P}_j)) \\ &+ \mathrm{Log}(\det(\mathbf{I}_n - \mathbf{C}^\top\mathbf{C})) \\ &- \mathrm{Log}(\det(\mathbf{P}_i + \mathbf{P}_j - \sqrt{\mathbf{P}_i}^\top\mathbf{C}\sqrt{\mathbf{P}_j} - \sqrt{\mathbf{P}_j}^\top\mathbf{C}^\top\sqrt{\mathbf{P}_i})),\end{aligned} \tag{33}$$

which enables us to write the optimization problem (19) in the equivalent form (26) given in the statement. This completes our proof. □

Next, we point out an interesting relationship between our estimate of the cross-covariance matrix in MAC and an estimate that one can obtain by generating sigma points around the means and applying a method similar to the one used in UKF method [14, 15]. Following the UKF approach, $2n+1$ sigma points for the sensor station s_r, $r = \{i, j\}$ are given as

$$\mathbf{x}_r^{(0)} = \hat{\mathbf{x}}_r, \tag{34a}$$

$$\mathbf{x}_r^{(\ell)} = \hat{\mathbf{x}}_r + \tilde{\mathbf{x}}_r^{(\ell)}, \qquad \ell = 1, \ldots, 2n, \tag{34b}$$

$$\tilde{\mathbf{x}}_r^{(\ell)} = \begin{cases} \left(\sqrt{(n+\lambda)\mathbf{P}_r}\right)_\ell^\top, & \ell = 1, \ldots, n, \\ -\left(\sqrt{(n+\lambda)\mathbf{P}_r}\right)_\ell^\top, & \ell = n+1, \ldots, 2n, \end{cases} \tag{34c}$$

where $\left(\sqrt{(n+\lambda)\mathbf{P}_r}\right)_\ell^\top$ is the ℓ-th row of matrix square root of $(n+\lambda)\mathbf{P}_r$. The weighting coefficients are defined as

$$w^{(0)} = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta, \tag{35a}$$

$$w^{(\ell)} = \frac{1}{2(n+\lambda)}, \qquad \ell = 1, \ldots, 2n, \tag{35b}$$

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 follows

$$\mathbf{P}_{ij} = \sum_{\ell=1}^{2n} w^{(\ell)} \mathbf{E}[\left(\hat{\mathbf{x}}_i - \mathbf{x}_i^{(\ell)}\right)\left(\hat{\mathbf{x}}_j - \mathbf{x}_j^{(\ell)}\right)^\top]. \tag{36}$$

Since $\mathbf{x}_r^{(\ell)} = -\mathbf{x}_r^{(\ell+n)}$, $r = \{i, j\}$, equation (36) is reduced to

$$\mathbf{P}_{ij} = 2\sum_{\ell=1}^{n} w^{(\ell)} \mathbf{E}[\left(\hat{\mathbf{x}}_i - \mathbf{x}_i^{(\ell)}\right)\left(\hat{\mathbf{x}}_j - \mathbf{x}_j^{(\ell)}\right)^\top]. \tag{37}$$

Substituting (34b) and (35b) into (37), yields

$$\mathbf{P}_{ij} = \frac{2}{2(n+\lambda)} \sum_{\ell=1}^{n} \big(\sqrt{(n+\lambda)\mathbf{P}_i}\big)_\ell^\top \big(\sqrt{(n+\lambda)\mathbf{P}_j}\big)_\ell,$$

which gives us the estimated cross-covariance by sigma points as follows

$$\mathbf{P}_{ij} = \sum_{\ell=1}^{n} \big(\sqrt{\mathbf{P}_i}\big)_\ell^\top \big(\sqrt{\mathbf{P}_j}\big)_\ell = \sqrt{\mathbf{P}_i}^\top \sqrt{\mathbf{P}_j}.$$

This cross-covariance guarantees the positive semi-definiteness of joint covariance matrix but it over estimates the correlation of the estimates.

We close this section with few remarks regarding the consistency analysis of MAC. Given a process with random state $\mathbf{x}$, a state estimator of this process which produces estimate $\hat{\mathbf{x}}$ with associated error covariance $\mathbf{P}$ is said to be consistent if it is unbiased, i.e., $\mathbf{E}[\mathbf{x}-\hat{\mathbf{x}}] = \mathbf{0}_{n\times 1}$ and its state estimation error satisfies $\mathbf{P} = \mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top]$ (see e.g., [23, 24, 19]). The definition of estimator consistency sometimes is relaxed from covariance matching to $\mathbf{P} \geq \mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top]$ (see e.g., [8] or other covariance intersection method literature). In Theorem 4.1, we showed 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 (22), with appropriate manipulations, we can rewrite $\mathbf{P}_{\text{MAC}}$ in (21b) as

$$\mathbf{P}_{\text{MAC}} = \mathbf{W}_i\mathbf{P}_i\mathbf{W}_i^\top + \mathbf{W}_i\mathbf{X}^\star\mathbf{W}_j^\top + \mathbf{W}_j\mathbf{X}^{\star\top}\mathbf{W}_i^\top + \mathbf{W}_j\mathbf{P}_j\mathbf{W}_j^\top.$$

Using the same gains, we can also obtain (recall $\mathbf{W}_i + \mathbf{W}_j = \mathbf{I}_n$)

$$\begin{aligned}\mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top] &= \mathbf{E}[\big(\mathbf{W}_i(\mathbf{x}-\hat{\mathbf{x}}_i) + \mathbf{W}_j(\mathbf{x}-\hat{\mathbf{x}}_j)\big)\big(\mathbf{W}_i(\mathbf{x}-\hat{\mathbf{x}}_i) + \mathbf{W}_j(\mathbf{x}-\hat{\mathbf{x}}_j)\big)^\top] \\ &= \mathbf{W}_i\mathbf{P}_i\mathbf{W}_i^\top + \mathbf{W}_i\mathbf{P}_{ij}\mathbf{W}_j^\top + \mathbf{W}_j\mathbf{P}_{ij}^\top\mathbf{W}_i^\top + \mathbf{W}_j\mathbf{P}_j\mathbf{W}_j^\top. \end{aligned} \tag{38}$$

As a result we can write $\mathbf{P}_{\text{MAC}} - \mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top] = \mathbf{W}\,\mathbf{M}\,\mathbf{W}^\top$, where

$$\mathbf{W} \triangleq \begin{bmatrix}\mathbf{W}_i & \mathbf{W}_j\end{bmatrix}, \quad \mathbf{M} \triangleq \begin{bmatrix}\mathbf{0}_n & \mathbf{X}^\star - \mathbf{P}_{ij} \\ (\mathbf{X}^\star - \mathbf{P}_{ij})^\top & \mathbf{0}_n\end{bmatrix}.$$

One can show that 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 with their set of magnitudes being equal to the set of singular values of $\mathbf{X}^\star - \mathbf{P}_{ij}$ (c.f.[25]). 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 $\mathbf{X}^\star - \mathbf{P}_{ij}$, $\mathbf{P}_{\text{MAC}} - \mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top] = \mathbf{W}\,\mathbf{M}\,\mathbf{W}^\top$ can be positive (semi-) definite, negative (semi-) definite or indefinite. Three numerical examples are given in Figure 2 that compare 1σ error ellipsoid of $\mathbf{P}_{\text{MAC}}$ with the one corresponding to $\mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$. As depicted in this figure, 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 are also measured by statistical consistency tests such as the Average Normalised Estimation Error Squared (ANEES) [24]. A statistical consistency test is based on the results of the Monte Carlo simulations that provide M independent samples $\mathbf{e}_i(k)$, $i \in \{1, \cdots, M\}$, of the estimation error $\mathbf{e}_i(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ where $\mathbf{x}(k) \in \mathbb{R}^n$ is the true state and $\hat{\mathbf{x}}(k) \in \mathbb{R}^n$ is the filter estimate with the associated error covariance $\mathbf{P}(k) \in \mathbb{S}^n_{++}$. In the ANEES test, we first obtain the Normalised Estimation Error Squared (NEES) via

$$\epsilon_M(k) = \frac{1}{M} \sum_{i=1}^{M} \mathbf{e}_i^\top(k) \mathbf{P}^{-1}(k) \mathbf{e}_i(k). \tag{39}$$

The random variable $\epsilon_M(k)$ has chi-squared distribution with n degree of freedom. Then, the ANEES consistency measure is obtained as

$$\bar{\epsilon}(k) = \frac{1}{n} \epsilon_M(k). \tag{40}$$

An estimation is consistent, if and only if the value of ANEES measure $\bar{\epsilon}(k)$ is close to 1. If ANEES is much greater than 1, the actual estimation error is much larger than what the estimator believes i.e., the estimator is too optimistic; if ANEES is much smaller than 1, the actual estimation error is much smaller than what the estimator believes i.e., the estimator is too pessimistic [26]. A simulation study demonstrating the ANEES measure for MAC fusion is presented in Section 5. As shown in Figure 3, for this example MAC produces consistent results (for further details see Section 5).

5. Simulation

In order to demonstrate the performance of our proposed MAC track-to-track fusion algorithm in comparison to the optimal fusion and also the covariance intersection method, a target tracking problem is considered. In the optimal fusion algorithm (indicated by OPT on the figures), the exact cross-covariances are maintained and propagated via (10). The target is assumed to be moving in one dimensional space with its state vector $\mathbf{x}$ being composed of target's position ξ and linear velocity $\dot{\xi}$, i.e., $\mathbf{x} = [\, \xi \quad \dot{\xi} \,]^\top$. The target's motion is modeled by the following discrete-time equation

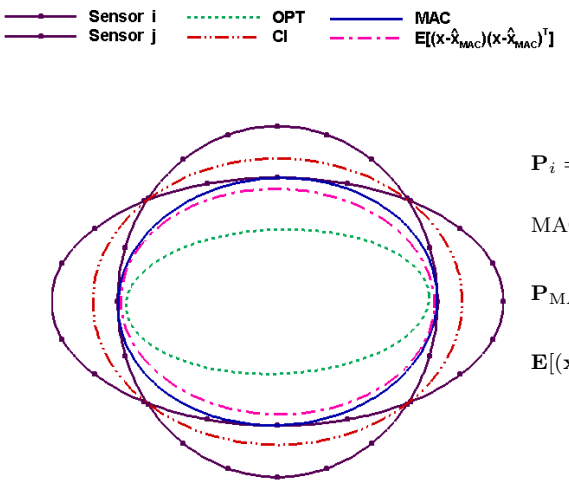
$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \omega(k).$$

The sampling time here is $\Delta t = 0.01\ s$ and the one dimensional stochastic process noise ω is a white Gaussian noise with covariance $\mathbf{Q} = 0.01$. Two sensors are tracking a common target, each with its local measurement model

$$\mathbf{z}_r(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_r(k) + \nu_r(k), \qquad r = \{i, j\}, \tag{41}$$

where the two measurement noises $\nu_i(k)$ and $\nu_j(k)$ are Gaussian zero mean processes with covariances $\mathbf{R}_i = 0.8$ and $\mathbf{R}_j = 1$, respectively. The two measurement noises are assumed to be mutually independent i.e., $\mathbf{E}[\nu_i(k)\nu_j^\top(k)] = \mathbf{0}_n$.

We use ANEES and RMSE tests as our performance measure. ANEES is used for consistency test and RMSE is used to evaluate the tracking error of the estimates. In our analysis, we use samples obtained from 1000 Monte-Carlo runs. Figure 3 demonstrates how the consistency provided by the MAC algorithm compares to the other fusion methods. As this figure shows, our three methods of interest are consistent which means, after convergence, the ANEES for each method remains close to 1. It is worth noticing that the proposed MAC method demonstrates an ANEES performance that is very close to the one obtained by the CI method. In this simulation scenario, the filters converge fast and after convergence the direction of eigenvectors of two local covariance matrices provided by sensor station s_i and s_j are approximately the same (see Figure 4). Therefore, both the MAC and the CI fusion rules choose the local covariance matrix with the smaller eigenvalues as the fused covariance matrix. From the geometric viewpoint, it implies when one error ellipsoid is completely inside the other one, the resulting estimates provided by the MAC and the CI is close to each other. However, when the eigenvectors of local covariance matrices at sensor stations s_i and s_j are not in the same direction, the MAC algorithm delivers tighter error ellipsoid than the CI method. Figure 5 shows the RMSE plot for our numerical example. As one can see, the proposed MAC algorithm has a lower RMSE than the CI method, especially in the initial time steps when the semi axes of error ellipsoids corresponding to the fused tracks are not parallel. This result also affirms that MAC provides unbiased estimates and after convergence, its accuracy is close to optimal fusion rule.



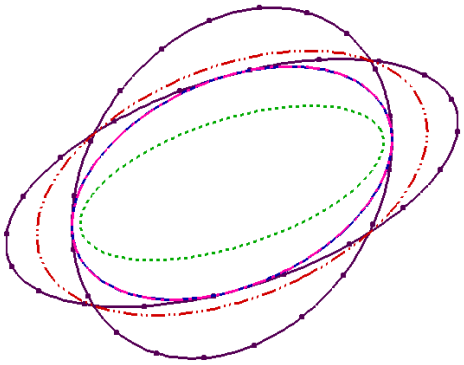
$$\mathbf{P}_i = \begin{bmatrix} 20 & 0 \\ 0 & 9 \end{bmatrix}, \ \mathbf{P}_j = \begin{bmatrix} 10 & 0 \\ 0 & 18 \end{bmatrix}, \ \mathbf{P}_{ij} = \begin{bmatrix} 5.657 & 0.949 \\ 0.474 & -6.364 \end{bmatrix}$$

$$\text{MAC's gains: } \mathbf{W}_i = \begin{bmatrix} 0.055 & -0.034 \\ 0.031 & 0.945 \end{bmatrix}, \ \mathbf{W}_j = \begin{bmatrix} 0.945 & 0.034 \\ -0.031 & 0.055 \end{bmatrix}$$

$$\mathbf{P}_{\text{MAC}} = \begin{bmatrix} 9.956 & 0.288 \\ 0.288 & 8.961 \end{bmatrix}, \ \mathbf{X}^\star = \begin{bmatrix} 9.398 & -0.649 \\ 0.324 & 8.458 \end{bmatrix}$$

$$\mathbf{E}[(\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}})^\top] = \begin{bmatrix} 9.599 & -0.121 \\ -0.121 & 7.432 \end{bmatrix}$$

(a)



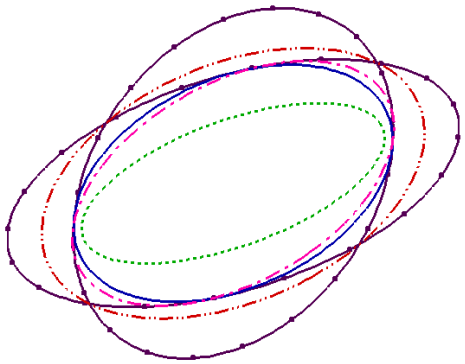
$$\mathbf{P}_i = \begin{bmatrix} 20 & 6.5 \\ 6.5 & 9 \end{bmatrix}, \ \mathbf{P}_j = \begin{bmatrix} 10 & 3.3 \\ 3.3 & 18 \end{bmatrix}, \ \mathbf{P}_{ij} = \begin{bmatrix} 5.657 & 1.867 \\ 1.839 & -4.789 \end{bmatrix}$$

$$\text{MAC's gains: } \mathbf{W}_i = \begin{bmatrix} -0.001 & 0.004 \\ -0.324 & 1.001 \end{bmatrix}, \ \mathbf{W}_j = \begin{bmatrix} 1.001 & -0.004 \\ 0.324 & -0.001 \end{bmatrix}$$

$$\mathbf{P}_{\text{MAC}} = \begin{bmatrix} 10.000 & 3.263 \\ 3.263 & 7.952 \end{bmatrix}, \ \mathbf{X}^\star = \begin{bmatrix} 9.999 & 3.273 \\ 3.263 & 7.955 \end{bmatrix}$$

$$\mathbf{E}[(\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}})^\top] = \begin{bmatrix} 10.000 & 3.261 \\ 3.261 & 7.952 \end{bmatrix}$$

(b)



$$\mathbf{P}_i = \begin{bmatrix} 20 & 6 \\ 6 & 9 \end{bmatrix}, \ \mathbf{P}_j = \begin{bmatrix} 10 & 4 \\ 4 & 18 \end{bmatrix}, \ \mathbf{P}_{ij} = \begin{bmatrix} 5.657 & 3.168 \\ 2.121 & -4.313 \end{bmatrix}$$

$$\text{MAC's gains: } \mathbf{W}_i = \begin{bmatrix} -0.021 & 0.077 \\ -0.278 & 1.021 \end{bmatrix}, \ \mathbf{W}_j = \begin{bmatrix} 1.021 & -0.077 \\ 0.278 & -0.021 \end{bmatrix}$$

$$\mathbf{P}_{\text{MAC}} = \begin{bmatrix} 9.944 & 3.263 \\ 3.263 & 8.255 \end{bmatrix}, \ \mathbf{X}^\star = \begin{bmatrix} 9.935 & 3.146 \\ 3.261 & 8.223 \end{bmatrix}$$

$$\mathbf{E}[(\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x} - \hat{\mathbf{x}}_{\text{MAC}})^\top] = \begin{bmatrix} 10.097 & 4.298 \\ 4.298 & 8.807 \end{bmatrix}$$

(c)

Figure 2 – 1σ ellipses of two correlated estimates, and their fused estimate computed by the the fusion algorithms MAC, CI and OPT (the Optimal filter (8)) for three different numerical examples. In all three cases, the optimal filter delivers the best result. The MAC and the CI methods are both conservative, but the MAC algorithm produces better results. The plots also depict how $\mathbf{P}_{\text{MAC}}$ compares to $\mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$ given in (38): plot (a) demonstrates an example in which $\mathbf{P}_{\text{MAC}}$ is larger than $\mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$; plot (b) demonstrates a case where $\mathbf{P}_{\text{MAC}}$ almost matches $\mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$, and plot (c) demonstrates a case that $\mathbf{P}_{\text{MAC}}$ intersects $\mathbf{E}[(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})(\mathbf{x}-\hat{\mathbf{x}}_{\text{MAC}})^\top]$.

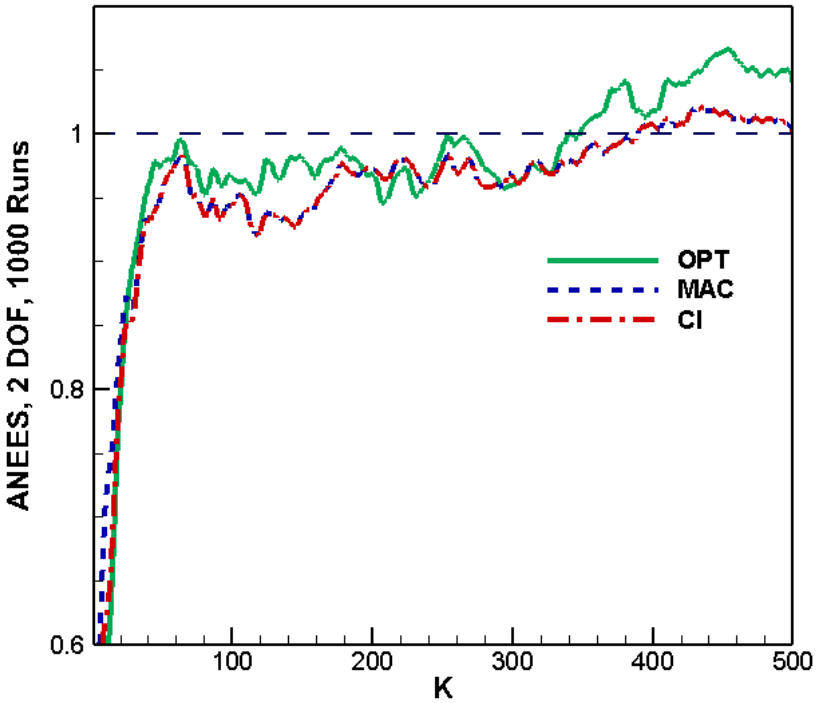
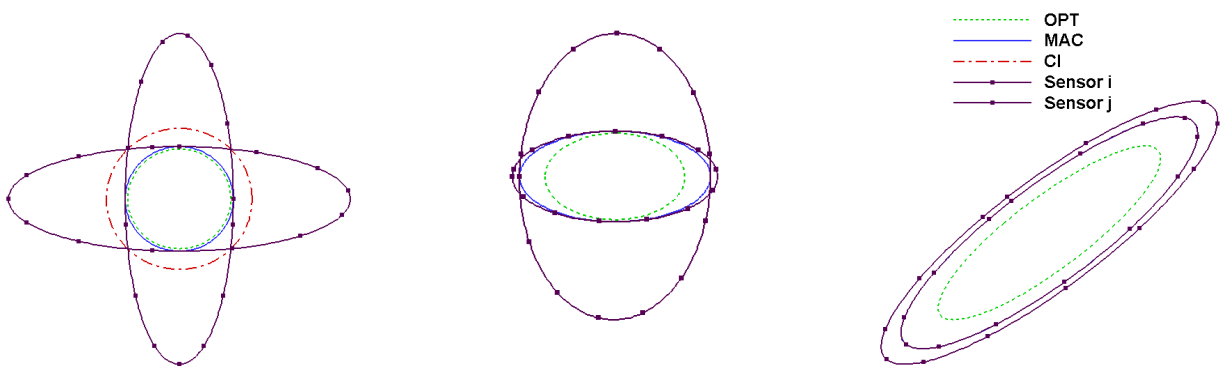


Figure 3 – The tracking ANEES by the MAC algorithm compared to the CI and OPT (optimal filter) fusion methods.



(a) time step k=1 (b) time step k=10 (c) time step k=100

Figure 4 – The 1σ error ellipses corresponding to the MAC, CI and OPT (optimal filter) fusion methods at three time steps $k = 1$, $k = 10$, and $k = 100$.

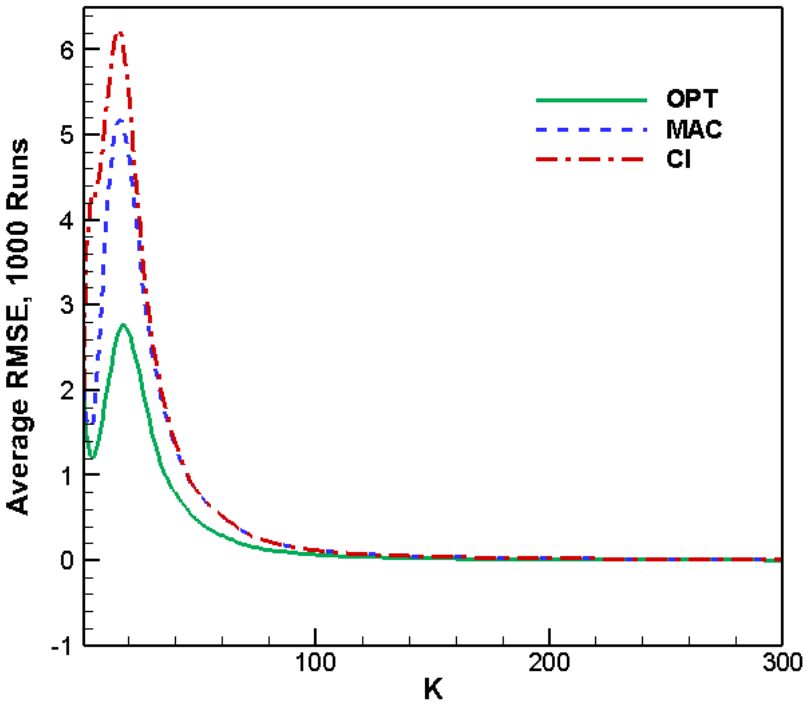


Figure 5 – The tracking average RMSE provided by the MAC algorithm compared to the CI and OPT (optimal filter) fusion methods.

6. Conclusions

This paper considered the problem of track-to-track fusion under unknown correlation between the fused tracks. We proposed a novel fusion rule that delivers less conservative estimates compared to the covariance intersection method while preserving the same nice loosely coupled feature of the covariance intersection method for track-to-track fusion. Our algorithm works based on estimating the unknown cross-covariance between the tracks in a way that the total uncertainty of the fused track is less than the one obtained by the covariance intersection method but is greater than that of the optimal track one obtains when the exact cross-covariance is known. We used rigorous analysis to study and demonstrate various properties of our proposed algorithm. We also analysed the covariance consistency of our algorithm and showed that our method may violate the covariance consistency condition in some occasions but not necessarily at all times. In numerical examples, we have observed that this violation is usually very small. We conclude that although the MAC algorithm does not necessarily provide rigorous theoretical guarantees for the covariance consistency, it is likely to perform better than the covariance intersection method in practical applications because of its less conservative joint covariance compared to that of the covariance intersection method. Our future work includes further research to empower the proposed algorithm to perform the optimization computations fast and more efficiently.

References

[1] J. K. Uhlmann, Covariance consistency methods for fault-tolerant distributed data fusion, Information Fusion 4 (3) (2003) 201–215.

[2] C. Y. Chong, S. Mori, K. C. Chang, W. H. Barker, Architectures and algorithms for track association and fusion, IEEE Transactions on Aerospace and Electronic Systems 15 (1) (2000) 5–13.

[3] E. Waltz, J. Llinas, Multisensor Data Fusion, Artech House, Norwood, MA, 1990.

[4] Y. Bar-Shalom, On the track-to-track correlation problem, IEEE Transactions on Automatic Control 26 (2) (1981) 571–572.

[5] R. A. Singer, A. J. Kanyuck, Computer control of multiple site track correlation, Automatica 7 (4) (1971) 455–463.

[6] P. S. Maybeck, Stochastic Models, Estimation, and Control, Academic Press, New York, 1982.

[7] B. Khaleghi, A. Khamis, F. O. Karray, S. N. Razavi, Multisensor data fusion: A review of the state-of-the-art, Information Fusion 14 (1) (2013) 28–44.

[8] S. J. Julier, J. K. Uhlmann, A non-divergent estimation algorithm in the presence of unknown correlations, in: American Control Conference, Vol. 4, 1997, pp. 2369–2373.

[9] A. Boulis, S. Ganeriwal, M. B. Srivastava, Aggregation in sensor networks: an energy–accuracy tradeoff, Ad Hoc Networks 1 (2–3) (2003) 317–331.

[10] S. J. Julier, J. K. Uhlmann, Using covariance intersection for slam, Robotics and Autonomous Systems 55 (1) (2007) 3–20.

[11] W. Qi, P. Zhang, Z. Deng, Robust sequential covariance intersection fusion Kalman filtering over multi-agent sensor networks with measurement delays and uncertain noise variances, Acta Automatica Sinica 40 (11) (2014) 2632–2642.

[12] G. Wang, N. Li, Y. Zhang, Diffusion distributed Kalman filter over sensor networks without exchanging raw measurements, Signal Processing 132 (2017) 1–7.

[13] R. Horst, N. V. Thoai, DC programming: Overview, Journal of Optimization Theory & Applications 103 (1) (1999) 1–43.

[14] S. J. Julier, J. K. Uhlmann, H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, IEEE Transactions on Automatic Control 45 (3) (2000) 477–482.
[15] S. J. Julier, J. K. Uhlmann, Unscented filtering and nonlinear estimation, Proceedings of the IEEE 92 (3) (2004) 401–422.
[16] R. A. Horn, C. R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1991.
[17] R. Horst, P. Pardalos, N. Thoai, Introduction to Global Optimization, Kluwer Academic Publishers, Dordrecht, 1995.
[18] Y. Bar-Shalom, L. Campo, The effect of the common process noise on the two-sensor fused-track covariance, IEEE Transactions on Aerospace and Electronic Systems 22 (6) (1986) 803–805.
[19] Y. Bar-Shalom, P. K. Willett, X. Tian, Tracking and Data Fusion: A Handbook of Algorithms, Yaakov Bar-Shalom, Storrs, CT, 2011.
[20] S. J. Julier, J. K. Uhlmann, Handbook of Data Fusion, CRC Press, 2001.
[21] J. K. Uhlmann, General data fusion for estimates with unknown cross covariances, in: Aerospace/Defense Sensing and Controls, Vol. 2755, 1996, pp. 536–547.
[22] R. C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, The International Journal of Robotics Research 5 (4) (1986) 56–68.
[23] H. Durrant-Whyte, Introduction to estimation and the Kalman filter, University of Sydney, Australian Centre for Field Robotics.
[24] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation, John Wiley & Sons, New York, 2001.
[25] R. C. Thompson, L. J. Freede, Eigenvalues of partitioned hermitian matrices, Bulletin of the Australian Mathematical Society 3 (01) (1970) 23–37.
[26] X. R. Li, Z. Zhao, 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, 2001, pp. 481–495.