

An Empirical Method to Fuse Partially Overlapping State Vectors for Distributed State Estimation

Joris Sijs¹, Uwe Hanebeck², and Benjamin Noack²

Abstract—State fusion is a method for merging multiple estimates of the same state into a single fused estimate. Dealing with multiple estimates is one of the main concerns in distributed state estimation, where an estimated value of the desired state vector is computed in each node of a networked system. Most solutions for distributed state estimation currently available assume that every node computes an estimate of the (same) global state vector. This assumption is impractical for systems observing large-area processes, due to the sheer size of the process state. A feasible solution is one where each node estimates a part of the global state vector, allowing different nodes in the network to have overlapping state elements. Although such an approach should be accompanied by a corresponding state fusion method, existing solutions cannot be employed as they merely consider fusion of two different estimates with equal state representations. Therefore, an empirical solution is presented for fusing two state estimates that have partially overlapping state elements. A justification of the proposed fusion method is presented, along with an illustrative case study for observing the temperature profile of a large rod, though a formal derivation is future research.

I. INTRODUCTION

State fusion is a method to merge multiple estimates of the same state vector into a single fused estimate. Such methods are beneficial in applications related to distributed state estimation. Typically, those applications consider a process that is being observed by multiple (local) subsystems, also called *nodes*, which are equipped with their own sensor, radio and CPU. Each node computes a local estimate of the process' state by combining its own measurement with data received from neighboring nodes, which could be either their local measurement or their local state estimation result (as computed by the neighbor). See, for example, solutions on distributed state estimation proposed in [9], [1], [8], [11] and the illustrative networked system in Figure 1.

The technology focus of this article will be on distributed state estimation, where nodes share their local state estimate, i.e., mean and covariance. A node can then improve its estimation result by merging its current local estimate with the estimates received from other nodes. Typically, merging of multiple state estimates is done via consensus methods, e.g., [17], [16], or via state fusion methods, e.g., [7], [4], [13]. The advantage of fusion solutions over consensus methods is that fusion will take the covariance (uncertainty) of the original estimates explicitly into account and thereby,

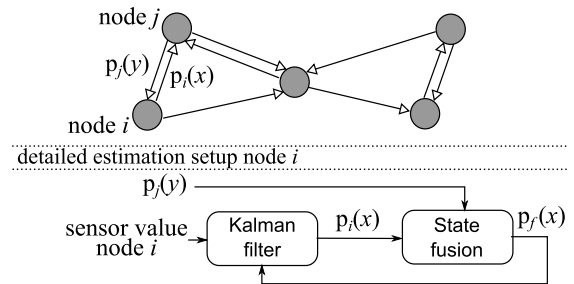


Fig. 1. Illustrative setup of a distributed Kalman filter, where each node i shares its local estimation result $p_i(x)$, as it is determined by its Kalman filtering algorithm. The detailed estimation scheme adopted by each node further indicates that local estimation results are fused with shared estimation results computed by neighboring nodes (in this example node j).

can reduce the covariance after fusion. However, where consensus methods are able to merge estimated means that have partially overlapping state elements, e.g., [15], fusion solutions require that the original estimates have an equal state representation. Note that, when distributed state estimation is considered, this latter statement implies that each node computes a local estimate of the *global* process' state vector. This requirement is undesirable, especially for large-area processes, such as climate, traffic, and the power grid, which have millions of state elements. Instead, a feasible approach is obtained by dividing the process into smaller subprocesses according to the spatial distribution of the nodes in the system (allowing overlap). Then, each node observes a local state vector containing a subset of the elements in the global state vector. Accordingly, solutions on decentralized control were established in [6], [14], though similar results on distributed state estimation based on fusion are still missing.

To solve this issue, a state fusion method is derived for merging two estimates with partly overlapping state elements. In addition, it is assumed that the cross-correlation of the original state vectors is unknown. This assumption is necessary as keeping track of cross-correlations between the different local estimates in a distributed state-estimator is impractical for a large-scale networked system. Further, for clarity of the proposed fusion solution, the considered estimates are characterized by Gaussian distributions.

II. PRELIMINARIES

\mathbb{R} , \mathbb{R}_+ , \mathbb{Z} , and \mathbb{Z}_+ define the set of real numbers, non-negative real numbers, integer numbers, and non-negative integer numbers, respectively. For any $\mathcal{C} \subset \mathbb{R}$, let $\mathbb{Z}_{\mathcal{C}} := \{c \in \mathbb{Z} | c \in \mathcal{C}\}$. The notation I_n and $0_{n \times m}$ are used to denote an

¹J. Sijs is with TNO Technical Sciences, Den Haag, The Netherlands, joris.sijs@tno.nl

²U. D. Hanebeck and B. Noack are with Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, uwe.hanebeck@kit.edu, benjamin.noack@kit.edu

$n \times n$ identity matrix and an $n \times m$ null-matrix (or -vector), respectively. The transpose and inverse of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted as A^\top and A^{-1} , respectively, while $\lambda_q(A)$ denotes the q -th eigenvalue of A . Further, $\{A\}_{i,j} \in \mathbb{R}$ denotes the element on the i -th row and j -th column of A . Given that $A, B \in \mathbb{R}^{n \times n}$ are positive definite, denoted with $A \succ 0$ and $B \succ 0$, then $A \succ B$ denotes $A - B \succ 0$.

Given the random-vectors $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, then the mean and auto-covariance of x are denoted as $\mathbb{E}[x]$ and $cov(x)$, respectively, while $cov(x, y)$ denotes the cross-covariance of x with y . More precisely, $cov(x) := \mathbb{E}[xx^\top] - \mathbb{E}[x]\mathbb{E}[x^\top]$ and $cov(x, y) := \mathbb{E}[xy^\top] - \mathbb{E}[x]\mathbb{E}[y^\top]$. Further, the terms *uncorrelated*, *correlated*, and *fully correlated* are introduced according to their definition proposed in [3].

Definition II.1 Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ be random vectors. Then, (i) x and y are *uncorrelated* if $\mathbb{E}[xy^\top] = \mathbb{E}[x]\mathbb{E}[y^\top]$; (ii) x and y are *correlated* if $y = Ax + z$ holds for some random vector $z \in \mathbb{R}^m$ uncorrelated with x and some $A \in \mathbb{R}^{m \times n}$; (iii) x and y are *fully correlated* if $y = Ax$ holds for some $A \in \mathbb{R}^{m \times n}$.

The Gaussian function (shortly noted as Gaussian) of vectors $x, \mu \in \mathbb{R}^n$ and matrix $\Sigma \in \mathbb{R}^{n \times n}$ is denoted as $G(x, \mu, \Sigma)$. If $G(x, \mu, \Sigma)$ is a probability density function (PDF) of a random vector x , then $\mu := \mathbb{E}[x]$ and $\Sigma := cov(x)$, where $\Sigma \succ 0$ is symmetric. Further, any Gaussian $G(x, \mu, \Sigma)$ can be represented by its (ellipsoidal) level-set $\mathcal{E}_{\mu, \Sigma} \subset \mathbb{R}^n$, which is defined as $\mathcal{E}_{\mu, \Sigma} := \{x | (x - \mu)^\top \Sigma^{-1} (x - \mu) \leq 1\}$.

III. PROBLEM FORMULATION

Let us consider two nodes i and j . Node i computes an estimate of the locally desired state $x \in \mathbb{R}^{n_x}$, i.e., $p_i(x)$, while node j computes an estimate of a (partially) different locally desired state $y \in \mathbb{R}^{n_y}$, i.e., $p_j(y)$. Both local estimates are described with a Gaussian distribution, for some suitable mean \hat{x}_i, \hat{y}_j and covariance X_i and Y_j , i.e.,

$$p_i(x) := G(x, \hat{x}_i, X_i) \quad \text{and} \quad p_j(y) := G(y, \hat{y}_j, Y_j).$$

To characterize the fact that x and y have partially overlapping state elements, let us introduce the three random vectors $u \in \mathbb{R}^{n_u}$, $v \in \mathbb{R}^{n_v}$, and $w \in \mathbb{R}^{n_w}$, having no overlapping state elements, such that the following holds:

$$x := \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{and} \quad y := \begin{pmatrix} u \\ w \end{pmatrix}. \quad (1)$$

Estimation results of $p_i(x)$ imply that node i has a Gaussian estimate of u and v locally, introduced as $p_i(u) = G(u, \hat{u}_i, U_i)$ and $p_i(v) = G(v, \hat{v}_i, V_i)$. Similarly, node j has a Gaussian estimate of u and w locally, introduced as $p_j(u) = G(u, \hat{u}_j, U_j)$ and $p_j(w) = G(w, \hat{w}_j, W_j)$. Then, for some suitable cross-correlations Q_i and R_j , one can observe that

$$\hat{x}_i = \begin{pmatrix} \hat{u}_i \\ \hat{v}_i \end{pmatrix}, \quad X_i = \begin{pmatrix} U_i & Q_i \\ Q_i^\top & V_i \end{pmatrix}, \quad (2a)$$

$$\hat{y}_j = \begin{pmatrix} \hat{u}_j \\ \hat{w}_j \end{pmatrix}, \quad Y_j = \begin{pmatrix} U_j & R_j \\ R_j^\top & W_j \end{pmatrix}. \quad (2b)$$

Therein, Q_i represents the cross-correlation of the random vectors u and v as determined by node i , while R_j represents the cross-correlation of u and w as determined by node j .

In the considered application of distributed state estimation node i and node j operate in a larger networked system. This means that although both nodes estimate u , note that node j could have determined $p_j(u)$ using information not available to node i . Hence, node i can still exploit unique information about u available in the estimate of node j , so to improve its own local estimate by fusing its current estimation result $p_i(x)$ with the estimate $p_j(y)$ of node j . Let us denote the estimation result obtained after fusing $p_i(x)$ with $p_j(y)$ as the (Gaussian) *fused* estimate $p_f(x) = G(x, \hat{x}_f, X_f)$, for some mean \hat{x}_f and some covariance X_f .

The main objective of this article is to find a fusion solution for merging two partially overlapping estimates $p_i(x)$ and $p_j(y)$ into a single fused estimate $p_f(x)$. To increase practical feasibility of the proposed solution for distributed state estimation one cannot assume that the cross-correlation of x and y is available, as this would impose strict requirements impractical for real applications. Existing fusion solutions further assume that the original state vectors have an equal state representation. Although they could be used for fusing the partial estimates $p_i(u)$ and $p_j(u)$, they do not consider the fact that improving the partial estimate of u in node i yields an improvement of the estimate of v as well (via the cross-correlation $Q_i = cov(u, v)$). To solve this issue, a novel fusion solution is proposed that will convert $p_j(y)$ into an estimate of x , after which standard fusion solutions for equal state vector representations can be employed.

IV. FUSION OF PARTIALLY OVERLAPPING STATES

The proposed solution for fusing $p_i(x)$ with $p_j(y)$ is presented in two steps. Firstly, it is assumed that there exists a solution for converting $p_j(y)$ into an estimate of x . This “converted” estimate is characterized by the Gaussian PDF $p_j(x) = G(x, \hat{x}_j, X_j)$, for some suitable mean \hat{x}_j and covariance X_j . Standard fusion solutions for merging two estimates $p_i(x)$ and $p_j(x)$ with an unknown cross-correlation can thus be employed, though two will be presented here. Secondly, the main contribution of this article is presented, which is a solution for an estimate of x based on information available to node j , i.e., converting $p_j(y)$ into $p_j(x)$.

A. State fusion with unknown correlations

The two estimates for fusion are both characterized by a Gaussian distribution of x , i.e.,

$$p_i(x) = G(x, \hat{x}_i, X_i) \quad \text{and} \quad p_j(x) = G(x, \hat{x}_j, X_j).$$

Numerous methods for fusing two estimates of the same state x are available. However, when cross-correlation between these estimates is not given, the remaining fusion methods still applicable are found in [7], [2], [18], [13]. Basically, these methods compute the fused estimate $p_f(x) = G(x, \hat{x}_f, X_f)$ of $p_i(x)$ and $p_j(x)$ by considering a worst case cross-correlation scenario. Two methods are presented here: a popular method known as “Covariance Intersection”, see

[7], and an improved fusion method resulting in a smaller covariance (uncertainty) after fusion known as ‘‘Ellipsoidal Intersection’’, see [12].

Covariance Intersection is an early developed state fusion method and still widely used today. The method determines the fused estimate as a convex combination of $p_i(x)$ and $p_j(x)$, for some $\omega \in \mathbb{R}_{[0,1]}$, and defines the fused mean \hat{x}_f and covariance X_f as follows:

$$\begin{aligned} X_f &:= (\omega X_i^{-1} + (1 - \omega)X_j^{-1})^{-1}, \\ \hat{x}_f &:= X_f(\omega X_i^{-1}\hat{x}_i + (1 - \omega)X_j^{-1}\hat{x}_j). \end{aligned} \quad (3)$$

Popularity of this fusion method led to numerous approaches for determining ω , see, e.g., [5], [2], [4], [10]. However, a drawback of Covariance Intersection is that the uncertainty after fusion does not decrease, i.e., $X_f \preceq X_i$ and $X_f \preceq X_j$ are not satisfied. This issue was solved by Ellipsoidal Intersection, which is presented, next.

Ellipsoidal Intersection starts by finding an explicit expression of the (unknown) cross-correlation, which is then used for merging *independent parts* of $p_i(x)$ and $p_j(x)$ via algebraic fusion formulas. The (unknown) cross-correlation is part of a newly introduced Gaussian estimate based on information that $p_i(x)$ and $p_j(x)$ share (mutual). As such, let us introduce a *mutual covariance* $\Gamma \in \mathbb{R}^{n_x \times n_x}$ and a *mutual mean* $\gamma \in \mathbb{R}^{n_x}$. With these newly introduced mutual variables, the algebraic fusion formulas for finding the fused mean \hat{x}_f and covariance X_f , yield

$$\begin{aligned} X_f &= (X_i^{-1} + X_j^{-1} - \Gamma^{-1})^{-1}, \\ \hat{x}_f &= X_f(X_i^{-1}\hat{x}_i + X_j^{-1}\hat{x}_j - \Gamma^{-1}\gamma). \end{aligned} \quad (4)$$

The mutual mean γ and mutual covariance Γ are determined via an eigenvalue decomposition, i.e., $\Sigma = SDS^{-1}$, which is denoted as $[S, D] = \text{eig}(\Sigma)$ for a positive definite $\Sigma \in \mathbb{R}^{n \times n}$, a diagonal $D \in \mathbb{R}^{n \times n}$ and a rotation matrix $S \in \mathbb{R}^{n \times n}$. As such, let us introduce the matrices $D_i, D_j, S_i, S_j \in \mathbb{R}^{n \times n}$ via the eigenvalue decompositions

$$\begin{aligned} [S_i, D_i] &= \text{eig}(X_i) \text{ and} \\ [S_j, D_j] &= \text{eig}(D_i^{-\frac{1}{2}}S_i^{-1}X_jS_iD_i^{-\frac{1}{2}}). \end{aligned}$$

Then, explicit expressions of γ and Γ , for some $\zeta \in \mathbb{R}_+$, yield

$$\begin{aligned} D_\Gamma &= \text{diag}(\max[1, \{D_j\}_{1,1}], \dots, \max[1, \{D_j\}_{n,n}]), \\ \Gamma &= S_i D_i^{\frac{1}{2}} S_j D_\Gamma S_j^{-1} D_i^{\frac{1}{2}} S_i^{-1}, \\ \gamma &= (X_i^{-1} + X_j^{-1} - 2\Gamma^{-1} + 2\zeta I_{n_x})^{-1} \times \\ &\quad ((X_j^{-1} - \Gamma^{-1} + \zeta I_{n_x})\hat{x}_i + (X_i^{-1} - \Gamma^{-1} + \zeta I_{n_x})\hat{x}_j). \end{aligned}$$

A suitable value of ζ follows: $\zeta = 0$ if $|1 - \{D_j\}_{q,q}| > 10\epsilon$, for all $q \in \mathbb{Z}_{[1,n]}$ and some $\epsilon \in \mathbb{R}_{>0}$, while $\zeta = \epsilon$ otherwise. The design parameter ϵ supports a numerically stable result.

This completes a brief description of two state fusion methods that can be employed for fusing $p_i(x)$ with $p_j(x)$. Yet, note that *only* $p_i(x)$ is available to node i , while $p_j(x)$ is not and should be derived from the received estimate $p_j(y)$. Such a derivation is presented in the next section.

B. Converting an estimate of y into an estimate of x

Node i receives the estimation result $p_j(y)$ of another node j in the network. Note that this received PDF is an estimate of the local state vector y , while node i computes an estimate of a different state vector x , i.e., $p_i(x)$. Yet, these two local state vectors have partially overlapping elements represented by the random vector u . Section IV-A indicates that a fusion result $p_f(x)$ can be obtained from $p_i(x)$ and $p_j(y)$, when the Gaussian PDF $p_j(y)$ is converted into a corresponding Gaussian PDF of x introduced as $p_j(x)$. A solution for establishing the variables of $p_j(x)$ is presented, next, for which we recall the following notation:

$$\begin{aligned} p_j(x) &= G(x, \hat{x}_j, X_j), \quad x = \begin{pmatrix} u \\ v \end{pmatrix}, \\ p_j(y) &= G(y, \hat{y}_j, Y_j), \quad y = \begin{pmatrix} u \\ w \end{pmatrix}, \\ \text{where, } \hat{y}_j &= \begin{pmatrix} \hat{u}_j \\ \hat{w}_j \end{pmatrix}, \quad Y_j = \begin{pmatrix} U_j & R_j \\ R_j^T & W_j \end{pmatrix}. \end{aligned} \quad (5)$$

This section derives explicit expressions for the estimation variables \hat{x}_j and X_j based on the available PDF $p_j(y)$. The main idea of the proposed solution is to exploit a property of the previously presented fusion methods: coping with an unknown correlation of $p_i(x)$ and $p_j(x)$ is done by assuming the ‘‘maximum’’ correlation possible. When considering the converted $p_j(x)$, then this property allows that any information about x not available in $p_j(y)$, i.e., the state elements collected in v , can be ‘‘copied’’ from $p_i(x)$. As such, the following reasoning for finding $p_j(x)$ is followed:

Principle IV.1 The converted $p_j(x)$ and its original $p_j(y)$ are *fully* correlated regarding their shared random vector u , while, simultaneously, $p_j(x)$ and $p_i(x)$ are *fully* correlated with respect to their shared random vector v .

1) *A solution for the converted mean:* The above principle is instrumental in finding a solution for the converted PDF $p_j(x)$. Let us start by deriving an expression of converted mean \hat{x}_j in (5).

Lemma IV.2 Let $p_i(x)$ and $p_j(y)$ of (2) be given. Further, let $p_j(x)$ of (5) be constructed from $p_j(y)$ and $p_i(x)$ according to Principle IV.1. Then, it holds that

$$\hat{x}_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix}.$$

Proof: Let us introduce $u_{ii}, u_{ij}, u_{jj} \in \mathbb{R}^{n_u}$, $v_{ii}, v_{ij} \in \mathbb{R}^{n_v}$ and $w_{jj} \in \mathbb{R}^{n_w}$, such that $p_i(x) = p\left(\begin{pmatrix} u_{ii} \\ v_{ii} \end{pmatrix}\right)$, $p_j(x) = p\left(\begin{pmatrix} u_{ij} \\ v_{ij} \end{pmatrix}\right)$ and $p_j(y) = p\left(\begin{pmatrix} u_{ij} \\ w_{jj} \end{pmatrix}\right)$. Then, the lemma is proven in case both $\mathbb{E}[u_{ij}] = \mathbb{E}[u_{jj}]$ and $\mathbb{E}[v_{ij}] = \mathbb{E}[v_{ii}]$ hold. Applying the property that $p_j(x)$ and $p_j(y)$ are fully correlated with respect to their shared random vector u (see Principle IV.1) implies that $u_{ij} = I_{n_u} u_{jj}$ (see Definition II.1) and thus $\mathbb{E}[u_{ij}] = \mathbb{E}[u_{jj}]$. The statement that $\mathbb{E}[v_{ij}] = \mathbb{E}[v_{ii}]$ can be proven in a similar way from the property that $p_j(x)$ and $p_i(x)$ are fully correlated with respect to their shared random vector v , which completes this proof. ■

2) *A solution for the converted covariance:* Next, an expression is derived for the remaining variable X_j in (5). In line with the reasoning of Lemma IV.2, one could employ the straightforward characterization $X_j = \begin{pmatrix} U_j & Z \\ Z^\top & V_j \end{pmatrix}$, for some suitable cross-covariance $Z \in \mathbb{R}^{n_u \times n_v}$. However, as U_j is obtained from $p_j(y)$ and V_j from $p_i(x)$, there is no clear solution on how to determine this cross-covariance Z .

Therefore, an alternative (empirical) approach is proposed for finding an expression of the converted covariance X_j . To that extent, let us introduce a transformation matrix $T \in \mathbb{R}^{n_x \times n_x}$ satisfying two items.

- After transformation, $p_i(Tx)$ has I_{n_x} as covariance, which further implies that the transformed estimation results of u and v according to $p_i(x)$ are uncorrelated;
- $T \begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix} T^\top$ results in $\begin{pmatrix} \tilde{U}_j & 0 \\ 0 & 0 \end{pmatrix}$, where $U_j, \tilde{U}_j \in \mathbb{R}^{n_u}$, implying that the covariance $\text{cov}(u)$ determined by $p_j(y)$ is entirely described by \tilde{U}_j after the transformation.

More precisely, let us introduce the transformed state

$$\tilde{x} := Tx \quad \text{with} \quad \tilde{x} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix}, \quad (6)$$

for some $\tilde{u} \in \mathbb{R}^{n_u}$ and $\tilde{v} \in \mathbb{R}^{n_v}$. Then, the following assumption, which will be satisfied later this section, should hold:

Assumption IV.3 There exists a transformation $T \in \mathbb{R}^{n_x \times n_x}$ of full rank, such that $TX_iT^\top = I_{n_x}$ and $T \begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix} T^\top = \begin{pmatrix} \tilde{U}_j & 0 \\ 0 & 0 \end{pmatrix}$.

Next, the same transformation is applied on X_j for finding a solution of this converted covariance X_j via its transformed covariance TX_jT^\top . To that extent, let us recall the two main rules of Principle IV.1 characterizing the converted PDF $p_j(x)$ and translate their implication to TX_jT^\top as the covariance of $p_j(\tilde{x})$, i.e.,

Rule 1: The converted $p_j(x)$ and original $p_i(x)$ are *fully correlated* regarding their shared random vector v ;

Impl. Rule 1 implies that $p_j(\tilde{x})$ should be *fully correlated* with $p_i(\tilde{x})$ on those parts that correspond to the “pre-transformed” random vector v . Note that this rule will be satisfied when $p_j(\tilde{x})$ and $p_i(\tilde{x})$ are fully correlated on all parts, which further gives the following result for their covariance matrices:

$$TX_jT^\top = TX_iT^\top = I_{n_x}. \quad (7)$$

Rule 2 The converted $p_j(x)$ and its original $p_j(y)$ are *fully correlated* regarding their shared random vector u ;

Impl. Rule 2 implies that $p_j(\tilde{x})$ should be *fully correlated* with the “pre-transformed” $p_j(y)$ on those parts that correspond to the “pre-transformed” random vector u . Note that $\begin{pmatrix} \tilde{U}_j & 0 \\ 0 & 0 \end{pmatrix}$ introduced in Assumption IV.3 yields a covariance matrix where \tilde{U}_j is the representation of U_j after transformation, while U_j is the covariance of the “pre-transformed” u as determined by the original $p_j(y)$. Hence, satisfying this second rule by $p_j(\tilde{x})$ implies that its corresponding covariance TX_jT^\top is similar to $\begin{pmatrix} \tilde{U}_j & \tilde{Z}_j \\ \tilde{Z}_j^\top & \tilde{V}_j \end{pmatrix}$, for some

$\tilde{Z}_j \in \mathbb{R}^{n_u \times n_v}$ and $\tilde{V}_j \in \mathbb{R}^{n_v \times n_v}$. The value of \tilde{Z}_j and \tilde{V}_j is found in the implication for satisfying Rule 1, i.e., in (7), yielding $\tilde{Z}_j = 0_{n_u \times n_v}$ and $V_j = I_{n_v}$ and thus

$$TX_jT^\top = \begin{pmatrix} \tilde{U}_j & 0_{n_u \times n_v} \\ 0_{n_v \times n_u} & I_{n_v} \end{pmatrix}. \quad (8)$$

The above solution on the transformed covariance TX_jT^\top is based on the two rules stated in Principle IV.1. Hence, the result in (8) can be used to derive the following characterization of the converted covariance

$$X_j := T^{-1} \begin{pmatrix} \tilde{U}_j & 0_{n_u \times n_v} \\ 0_{n_v \times n_u} & I_{n_v} \end{pmatrix} T^{-\top}. \quad (9)$$

A solution for the transformation matrix T is presented in the next lemma, for which the rotation matrices $S_i, \tilde{S}_j \in \mathbb{R}^{n_x}$ and the diagonal matrices $D_i, \tilde{D}_j \in \mathbb{R}^{n_x}$ are introduced via a standard and an alternative eigenvalue decomposition¹:

$$[S_i, D_i] = \text{eig}(X_i), \\ [\tilde{S}_{U_j}, \tilde{D}_{U_j}] = \text{eig}^* \left(D_i^{-\frac{1}{2}} S_i^{-1} \begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix} S_i^{-\top} D_i^{-\frac{1}{2}} \right). \quad (10)$$

Lemma IV.4 Let the covariance matrices $X_i \in \mathbb{R}^{n_x}$ and $U_j \in \mathbb{R}^{n_u}$ be given and let $S_i, \tilde{S}_{U_j}, D_i$ and \tilde{D}_{U_j} suit (10). Then, the transformation matrix satisfying Assumption IV.3, yields

$$T = \tilde{S}_{U_j}^{-1} D_i^{-\frac{1}{2}} S_i^{-1}. \quad (11)$$

Proof: The proof is presented by substituting T of (11) and verify (i) $TX_iT^\top = I_{n_x}$ and (ii) $T \begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix} T^\top = \begin{pmatrix} \tilde{U}_j & 0 \\ 0 & 0 \end{pmatrix}$.

(i) Since S_i and D_i are the result from the eigenvalue decomposition $\text{eig}(X_i)$, i.e., $X_i = S_i D_i S_i^{-1} = S_i D_i S_i^\top$, one obtains that $D_i^{-\frac{1}{2}} S_i^{-1} X_i S_i^{-\top} D_i^{-\frac{1}{2}} = I_{n_x}$. Further, since \tilde{S}_{U_j} is a rotation matrix, $\tilde{S}_{U_j}^{-1} D_i^{-\frac{1}{2}} S_i^{-1} X_i S_i^{-\top} D_i^{-\frac{1}{2}} \tilde{S}_{U_j}^{-\top} = I_{n_x}$ holds, which is equivalent to TX_iT^\top when T follows (11).

(ii) The alternative eigenvalue decomposition in (10), i.e., denoted with $\text{eig}^*(\cdot)$, gives that $T \begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix} T^\top = \tilde{D}_{U_j}$. Since $\text{eig}^*(\cdot)$ arranges the eigenvalues on the diagonal of \tilde{D}_j in descending order, see footnote, and since it was based on a transformation of $\begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix}$ with a full rank transformation matrix T , a total of n_v eigenvalues in \tilde{D}_{U_j} are zero. Or stated differently, each element in \tilde{D}_{U_j} is zero except for the first set of diagonal elements $\{\tilde{D}_{U_j}\}_{1,1}$ until $\{\tilde{D}_{U_j}\}_{n_u, n_u}$ and thus $\tilde{D}_{U_j} = T \begin{pmatrix} U_j & 0 \\ 0 & 0 \end{pmatrix} T^\top$ is of the following form: $\begin{pmatrix} \tilde{U}_j & 0 \\ 0 & 0 \end{pmatrix}$. ■

To summarize, let us present an example of how one could implement the proposed solution for finding the mean \hat{x}_j and covariance X_j of the converted PDF $p_j(x)$. The algorithm differs in computing X_j from the derivation above, since it assumes a general eigenvalue decomposition where eigenvalues are not necessarily obtained in descending order.

¹The alternative eigenvalue decomposition denoted as $[\tilde{S}, \tilde{D}] = \text{eig}^*(A)$, for some suitable A , computes its result in such a way that the diagonal elements $\{\tilde{D}\}_{1,1} \dots \{\tilde{D}\}_{n,n}$ correspond to the eigenvalues in *descending order*, i.e., $\{\tilde{D}\}_{1,1} = \lambda_{\max}(A)$ and $\{\tilde{D}\}_{n,n} = \lambda_{\min}(A)$.

```

 $[\hat{x}_j, X_j] = \text{stateConversion}(\hat{x}_i, X_i, \hat{y}_j, Y_j)$ 
% compute some initial transformation matrices
 $T_{u,y} = (I_{n_u} \ 0_{n_u \times n_w}); \quad T_{v,x} = (0_{n_v \times n_u} \ I_{n_v});$ 
 $[S_i, D_i] = \text{eig}(X_i);$ 
% compute the converted mean
 $\hat{u}_j = T_{u,y} \hat{y}_j; \quad \hat{v}_i = T_{v,x} \hat{x}_i;$ 
 $\hat{x}_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_i \end{pmatrix};$ 
% compute the converted covariance
 $[S_{U_j}, D_{U_j}] = \text{eig} \left( \begin{pmatrix} T_{u,y} Y_j T_{u,y}^\top & 0_{n_u \times n_v} \\ 0_{n_u \times n_v} & 0_{n_v \times n_v} \end{pmatrix} \right);$ 
 $\Upsilon = 0_{n_x \times n_x};$ 
 $\{\Upsilon\}_{q,q} = \begin{cases} 1 & \text{if } \{D_{U_j}\}_{q,q} = 0, \\ \{D_{U_j}\}_{q,q} & \text{else,} \end{cases} \quad \forall q \in \mathbb{Z}_{[1, n_x]}$ 
 $X_j = S_i D_i^{\frac{1}{2}} S_{U_j} \Upsilon S_{U_j}^{-1} D_i^{\frac{1}{2}} S_i^{-1}.$ 

```

The above algorithm converts the PDF of node j , i.e., $p_j(y)$, into an estimate of x , i.e., $p_j(x)$. The desired fused estimate $p_f(x)$ is then computed by merging $p_i(x)$ with the converted $p_j(x)$. In the illustrative example presented, next, Ellipsoidal Intersection is employed for fusing $p_i(x)$ and $p_j(x)$.

V. ILLUSTRATIVE EXAMPLE

The proposed fusion method for unequal but overlapping state vectors is assessed in a distributed Kalman filtering example. The benchmark process is heat conduction in a rod, see Figure 2, which starts at a temperature of 300 K before being heated and cooled at different positions. The network of 5 nodes aims to reconstruct the temperature profile.

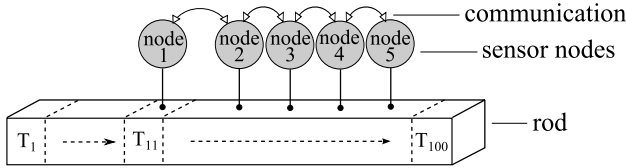


Fig. 2. An illustration of the considered setup for estimating the temperature profile of a rod via a network of 5 nodes.

The rod is divided into 100 segments and thus, the temperature T_q of each segment q is to be estimated. Details on specific values of the heat conduction process are omitted for brevity, though a mathematical representation of process dynamics is obtained by modeling the temperature value T_q at a sample instant $k+1$, for some $k \in \mathbb{Z}_+$, as follows:

$$T_q[k+1] = 0.17T_{q-1}[k] + 0.66T_q[k] + 0.17T_{q+1}[k] + d_q[k],$$

for all $q \in \mathbb{Z}_{[1,100]}$. The random variable $d_q[k]$ represents process noise and is introduced to model the fact that neither heating nor cooling values are available to the system. A suitable characterization of this random variable, yields

$$p(d_q[k]) = G(d_q[k], 0, 30) \quad \forall q \in \mathbb{Z}_{[1,100]}, \forall k \in \mathbb{Z}_+.$$

The rod is heated at segment 50 with 15 W and cooled with -10 W at segments 30 and 70. Further, both ends of the bar are stabilized at a temperature of 300 K. Nodes of the networked system are placed at segment 10, 30, 50, 70 and 90, where any two neighboring nodes share local data. Each node measures the temperature *value* of its own segment, while it estimates the temperature *profile* of a particular local area of the bar. More precisely, the local state vector θ_i and local measurement z_i , for all nodes $i \in \mathbb{Z}_{[1,5]}$, yield

$$\begin{aligned} \theta_1 &= (T_1, \dots, T_{40}), & z_1 &= T_{10} + s_1, \\ \theta_2 &= (T_1, \dots, T_{60}), & z_2 &= T_{30} + s_2, \\ \theta_3 &= (T_{20}, \dots, T_{80}), & z_3 &= T_{50} + s_3, \\ \theta_4 &= (T_{40}, \dots, T_{100}), & z_4 &= T_{70} + s_4, \\ \theta_5 &= (T_{60}, \dots, T_{100}), & z_5 &= T_{90} + s_5. \end{aligned}$$

The random variables s_i , for all $i \in \mathbb{Z}_{[1,5]}$, represent measurement noise and are stochastically characterized as follows:

$$p(s_i[k]) = G(s_i[k], 0, 0.1), \quad \forall i \in \mathbb{Z}_{[1,5]}, \forall k \in \mathbb{Z}_+.$$

Note that local state vectors are denoted as θ_i , for all $i \in \mathbb{Z}_+$ rather than x and y used to explain the proposed the state fusion solution. The main reason for adopting this notation is to reduce confusion, as the representation of x and y changes per node and per received estimate.

Two different types of distributed Kalman filtering approaches are assessed. A schematic setup of their estimation algorithm per node is depicted in Figure 3.

- LKF is an approach where nodes exchange their local measurements. As such, apart from its own measurement z_i , node i receives (at most) the local measurements z_{i-1} and z_{i+1} , which can be processed via a standard Kalman filtering algorithm for computing an estimate of localized state θ_i .
- DKF is the proposed approach in line with Figure 1. As such, neighboring nodes share their locally computed estimate, due to which node i receives (at most) estimates of θ_{i-1} and θ_{i+1} from the corresponding neighboring nodes. These are merged with its own estimate θ_i via the proposed fusion method to compute a fused estimate of the same localized state θ_i .

Setup of node i 's estimation algorithm

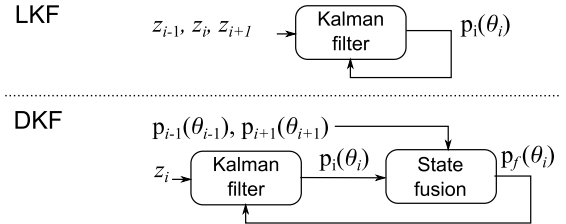


Fig. 3. An illustrative setup of the local algorithm performed by each node i according to the two distributed Kalman filtering approaches assessed. Note that neighboring measurements z_{i-1} and z_{i+1} and neighboring estimates $p_{i-1}(\theta_{i-1})$ and $p_{i+1}(\theta_{i+1})$ are only available to node i if these nodes exist. As such, node 1 will not receive z_0 neither $p_0(\theta_0)$, since there is no node 0.

Figure 4 and Figure 5 show the temperature profile as estimated by nodes 2 and 4 after 1200 s for both the LKF and the proposed DKF approach, respectively. Apart from a slight difference between the two approaches in estimation error, their main difference is in the agreement of the temperature value at those segments that are both estimated by node 2 and well as by node 4, i.e., segments 40 until 60. In case of the LKF, Figure 4 shows that node 2 and node 4 do not agree on the temperature value of these segments. This is due to the fact that neighboring nodes share their local measurement. As such, node 2 receives the temperature value at segments 10, 30, and 50 and is not able to reconstruct the decaying temperature *after* segment 50. Similarly, node 4 receives the temperature value at segments 50, 70, and 90 and is not able to reconstruct the decaying temperature *before* segment 50. This difference in the estimated temperature value of overlapping segments is solved in the DKF by exchanging local estimates (rather than local measurements) and employ the proposed fusion method. In that case, although node 4 has a measured temperature value from segment 70 only, it also receives the estimated temperature profile of node 3. This latter profile includes information on the temperature value of segment 30, since node 3 receives the local estimation result of node 2. It is exactly this progression of information from neighbor to neighbor to neighbor etc. that enables node 4 to reconstruct the decay in temperature before segment 50. Similarly, node 2 can reconstruct this decay as well.

Such benefits of the DKF approach would not be possible with an alternative fusion method than the one proposed in this article, as they require that each node computes a local estimate of the entire temperature profile, i.e., T_1 until T_{100} .

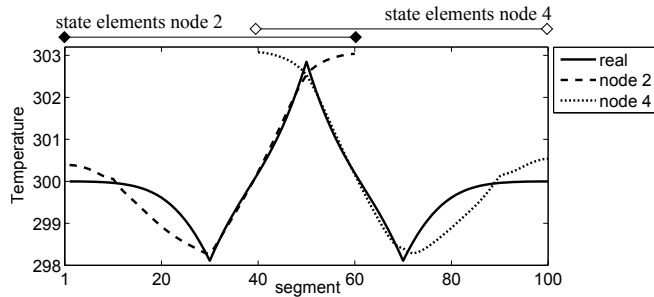


Fig. 4. The LKF results of the temperature profile estimated by node 2, i.e., between the segments 1 and 60, and the temperature profile estimated by node 4, i.e., between the segments 40 and 100.

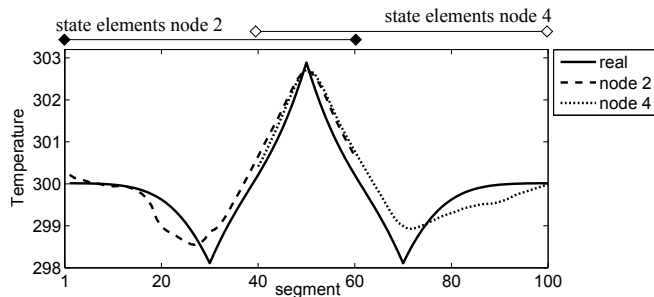


Fig. 5. The DKF results of the temperature profile estimated by node 2, i.e., between the segments 1 and 60, and the temperature profile estimated by node 4, i.e., between the segments 40 and 100.

VI. CONCLUSIONS

This article proposed a fusion solution for merging two different estimates that have overlapping state elements, while each estimate has a unique part as well. Existing fusion methods assume that the two different state estimates before fusion have the same state representation and could thus not be employed directly. Yet, to make use of these existing fusion methods, a solution was presented that turns one of the original estimates into a converted estimate of similar state representation as the other original estimate. An important assumption made for this conversion is that the employed fusion method is able to cope with unknown correlations. The main line of reasoning of the proposed fusion solution has been presented along with an empirical justification, which makes a mathematical argumentation of the fusion approach necessary future work.

REFERENCES

- [1] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26:622–633, 2008.
- [2] L. Chen, P.O. Arambel, and R.K. Mehra. Fusion under unknown correlation: Covariance intersection as a special case. In *5-th IEEE Int. Conf. on Information Fusion*, pages 905–912, 2002.
- [3] A. L. Edwards. *Multiple regression and the analysis of variance and covariance*. W. H. Freeman & Co Ltd, 1979.
- [4] D. Franken and A. Hupper. Improved fast covariance intersection for distributed data fusion. In *8-th Int. Conf. on Information Fusion*, pages WbA23:1–7, Philadelphia, PA, USA, 2005.
- [5] U. D. Hanebeck, K. Briechele, and J. Horn. A tight bound for the joint covariance of two random vectors with unknown but constrained cross-correlation. In *IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 85–90, Baden-Baden, Germany, 2001.
- [6] M. Ikeda and E. Siljak, D. D. Duranth-White. Decentralized control with overlapping information sets. *Journal of Optimization Theory and Applications*, 34:279–310, 1981.
- [7] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conf.*, pages 2369–2373, Piscataway, New Jersey, USA, 1997.
- [8] S.J. Julier. Estimating and exploiting the degree of independent information in distributed data fusion. In *12-th Int. Conf. on Information Fusion*, pages 772–779, Seattle, Washington, USA, 2009.
- [9] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *46-th IEEE Conf. on Decision and Control*, pages 5492 – 5498, New Orleans, Louisiana, USA, 2007.
- [10] M. Reinhardt, B. Noack, M. Baum, and U.W. Hanebeck. Analysis of set-theoretic and stochastic models for fusion under unknown correlations. In *In thr Proc. of the 14th Int. Conf. on Information Fusion*, Chicago, Illinois, USA, 2011.
- [11] J. Sijs. *State estimation in networked systems*. PhD thesis, Eindhoven University of Technology, 2012.
- [12] J. Sijs and M. Lazar. State fusion with unknown correlation: Ellipsoidal intersection. *Automatica*, 48:1874 – 1878, 2012.
- [13] J. Sijs, M. Lazar, and P.P.J. v.d. Bosch. State fusion with unknown correlation: Ellipsoidal intersection. In *American Control Conf.*, pages 3992 – 3997, Baltimore, USA, 2010.
- [14] D. D. Siljak. *Decentralized Control of Complex Systems*. New York, Academic Press, 1991.
- [15] S. S. Stankovic, Stankovic M. S., and D. M. Stipanovic. Consensus based overlapping decentralized estimator. *IEEE Trans. on Automatic Control*, 54:410–415, 2009.
- [16] A. Tahbaz Salehi and A. Jadbabaie. Consensus over ergodic stationary graph processes. *IEEE Trans. on Autom. Control*, 55:225–230, 2010.
- [17] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *4-th Int. Symp. on Information processing in sensor networks*, pages 63 – 70, LA, CA, USA, 2005.
- [18] Y. Zhuo and J. Li. Data fusion of unknown correlations using internal ellipsoidal approximations. In *17-th IFAC World Congress*, pages 2856–2860, Seoul, Korea, 2008.