Fully Decentralized Estimation Using Square-Root Decompositions

Susanne Radtke, **Benjamin Noack**, and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
susanne.radtke@kit.edu, benjamin.noack@ieee.org, uwe.hanebeck@ieee.org

Abstract

Networks consisting of several spatially distributed sensor nodes are useful in many applications. While distributed information processing can be more robust and flexible than centralized filtering, it requires careful consideration of dependencies between local state estimates. This paper proposes an algorithm to keep track of dependencies in decentralized systems where no dedicated fusion center is present. Specifically, it addresses double-counting of measurement information due to intermediate fusion results and correlations due to common process noise and common prior information. To limit the necessary amount of data, this paper introduces a method to partially bound correlations, leading to a more conservative fusion result than the optimal reconstruction while reducing the necessary amount of data. Simulation studies compare the performance and convergence rate of the proposed algorithm to other state-of-the-art methods.

Comment: Relation to Prior Versions of this Paper

This paper is an extended version of [1], which won the Best Paper Award in the general category during the 23rd Conference on Information Fusion. Sec. 2, 3, and 4 have been improved to provide more clarity. Further, Sec. 5.2 has been updated with an improved implementation of the previously used consensus algorithms, and the resulting implications are discussed.

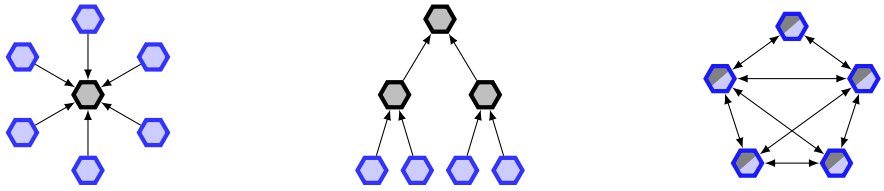
1. Introduction

Considered Problem: Sensor networks consist of several spatially distributed sensor nodes that can cooperatively perform a variety of different tasks [2], e.g., tracking a moving target using a network of cameras. In this paper, we consider the problem of fusing several state estimates in discrete-time linear Gaussian systems with multiple completely synchronized sensors with linear Gaussian observations. While centralized processing of measurements can be done optimally, network topology and communication bandwidth often forbid processing measurements in a central processing unit since nodes can only communicate with their closest neighbors. Distributed estimation allows the processing of measurements in a local processing unit. This local information is then communicated and fused with information from neighboring sensor nodes. It has been shown that the distributed processing of sensor data can be more robust, flexible, and scalable [3]. However, it introduces dependencies that need to be addressed carefully to ensure consistent fusion results.

State-of-the-Art: Within the past forty years, many algorithms [4] have been proposed to address the problems arising in distributed estimation. This includes using the information form of the Kalman filter [5]–[7] or formulating an optimally distributed Kalman filter [8]–[10]. Other approaches propose to use local Kalman filters and fuse their respective state estimates. Several publications address the correlations due to common process noise and common prior information [11]–[14]. When neglecting dependencies [15], fused estimates tend to become inconsistent as the uncertainty is underestimated. Covariance intersection [16]–[18] aims to find a conservative fusion rule to always ensure consistent results. As these are often too conservative, other approaches try to find closer bounds, e.g., inverse covariance intersection [19], [20]. Specifically for different network topologies, other algorithms such as the channel filter [3], the information graph approach [21] or the information matrix fusion [22], [23] were proposed.

Another class of algorithms aims to converge to a global estimate by iteratively exchanging information between neighboring nodes. Prominent representatives include consensus on measurements [24], consensus on information [25], [26], or hybrid approaches [27], [28]. Consensus methods can be regarded as suboptimal fusion rules [29] where the averaging of the information does not represent the actual information in the network and does also not consider redundant information

Manuscript received August 28, 2020; revised January 15, 2021; released for publication January 23, 2021.
Refereeing of this contribution was handled by Chee-Yee Chong.
This work was supported by the German Research Foundation (DFG) under grant HA 3789/14-1.
The authors are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, susanne.radtke@kit.edu, benjamin.noack@kit.edu, uwe.hanebeck@ieee.org.



(a) Centralized (b) Hierarchical (c) Decentralized

Fig. 1: Different network topologies with sensor nodes (blue), nodes only dedicated to fusion (gray) and sensor nodes with fusion capabilites (blue and gray).

systematically. For simpler network topologies, several approaches trying to reconstruct the cross-covariance matrix between state estimates using ensembles, e.g., the common past invariant Ensemble KF (CPI-EnKF) [30], or using samples [31]–[33] have been proposed. Furthermore, a reconstruction of cross-covariance matrices using square-root decompositions was proposed by [34], [35]. The reconstruction of cross-covariances has advantageous properties as it allows optimal fusion with consistent fusion results that are generally more accurate and do not over- or underestimate the uncertainty. Yet, it requires the communication of additional information between sensor nodes leading to a trade-off between optimality and network capacity.

Contribution: The square-root decomposition as initially proposed in [35] considers fusion in network topologies with only one dedicated fusion center. In this paper, we apply the decompositions to decentralized estimation tasks, where each node may sporadically serve as a fusion center. Nodes can exchange their estimates and fuse their local estimates with the received information. For this purpose, each node must keep track of correlations during its local processing steps. Not only common process noise needs to be encoded in the square-root decompositions, but also double counting of information poses a problem in decentralized network topologies and needs to be tracked. Due to the storage requirements and communication load associated with the square-root decompositions, the nodes can reach a compromise between fusion quality and resource demands by introducing partial bounds on the correlations.

Outline: The paper is structured as follows. In Sec. 2, we first discuss the problem of fusing several state estimates with correlated estimation errors. In Sec. 3, we revisit the previously proposed square-root decomposition method [35] to reconstruct the cross-covariance matrix between estimates in centralized sensor networks with only one dedicated fusion center. Decentralized network topologies in the absence of a dedicated fusion center are studied in Sec. 4. The evaluation in Sec. 5 studies different scenarios and also provides a comparison with consensus methods. Sec. 7 concludes the paper.

2. Problem Formulation

We consider a discrete-time linear time-variant stochastic dynamic system with time index k and state transition matrix $\mathbf{A}_k$, state vector $\underline{\boldsymbol{x}}_k \in \mathbb{R}^N$ of state dimension N, and zero-mean white Gaussian system noise $\underline{\boldsymbol{w}}_k$ with noise dimension $W = N$ and covariance matrix $\mathbf{Q}_k$, i.e.,

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}_k\, \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{w}}_k\,, \text{with}\ \ \underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q}_k)\,. \tag{1}$$

The system is observed by a network of N_s sensor nodes. The processing and sensing times of the sensor nodes are synchronized. Each individual node i receives measurements using the observation model $\mathbf{C}^i$ and covariance $\mathbf{R}_k^i$ according to

$$\underline{\boldsymbol{z}}_k^i = \mathbf{C}^i\, \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k^i\,, \text{with}\ \ \underline{\boldsymbol{v}}_k^i \sim \mathcal{N}(\underline{0}, \mathbf{R}_k^i)\,. \tag{2}$$

Further, we assume that the measurement noise and the process noise are mutually independent. Each node i computes a state estimate $\underline{\hat{x}}_{k|k}^i$ with error covariance matrix $\mathbf{P}_{k|k}^i = E\big[(\underline{\hat{x}}_{k|k}^i - \underline{\boldsymbol{x}}_k)(\underline{\hat{x}}_{k|k}^i - \underline{\boldsymbol{x}}_k)^{\mathrm{T}}\big]$.

2.1. Fusion of Estimates

Without loss of generality, we confine ourselves to the fusion of two estimates as multiple estimates can be fused sequentially. In the following discussions, we also omit the time index k for the sake of clarity.

The fusion of two state estimates $\underline{\hat{x}}^i$ and $\underline{\hat{x}}^j$ can take place at an arbitrary time step k and is a linear combination with the fusion gains $\mathbf{F}^i$ and $\mathbf{F}^j$. Depending on the chosen fusion algorithm, the gains can be determined according to the Bar-Shalom/Campo formulas but can also be fixed weighting matrices. The fused estimate becomes

$$\underline{\hat{x}}^{\mathrm{f}} = \mathbf{F}^i\, \underline{\hat{x}}^i + \mathbf{F}^j\, \underline{\hat{x}}^j\,, \tag{3}$$

with $\mathbf{F}^i + \mathbf{F}^j = \mathbf{I}$ and the corresponding error covariance matrix

$$\mathbf{P}^{\mathrm{f}} = \mathbf{F}^i \mathbf{P}^i (\mathbf{F}^i)^{\mathrm{T}} + \mathbf{F}^i \mathbf{P}^{i,j} (\mathbf{F}^j)^{\mathrm{T}} + \mathbf{F}^j \mathbf{P}^{j,i} (\mathbf{F}^i)^{\mathrm{T}} + \mathbf{F}^j \mathbf{P}^j (\mathbf{F}^j)^{\mathrm{T}} = \begin{bmatrix} \mathbf{F}^i & \mathbf{F}^j \end{bmatrix} \mathbf{J} \begin{bmatrix} \mathbf{F}^i & \mathbf{F}^j \end{bmatrix}^{\mathrm{T}}\,. \tag{4}$$

The joint error covariance matrix is

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^{i} & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^{j} \end{bmatrix},$$

where $\mathbf{P}^{i,j} = E\big[(\hat{\underline{x}}^{i} - \underline{\boldsymbol{x}})(\hat{\underline{x}}^{j} - \underline{\boldsymbol{x}})^{\mathrm{T}}\big] = \big(\mathbf{P}^{j,i}\big)^{\mathrm{T}}$ denote the cross-covariances and characterize the correlated estimation errors between the state estimates. Typically, the fusion gains $\mathbf{F}^{i}$ and $\mathbf{F}^{j}$ are computed to minimize the estimation error $E\big[(\hat{\underline{x}}^{\mathrm{f}} - \underline{\boldsymbol{x}})^{\mathrm{T}}(\hat{\underline{x}}^{\mathrm{f}} - \underline{\boldsymbol{x}})\big]$. In this case, we refer to $\hat{\underline{x}}^{\mathrm{f}}$ as the optimal fusion result. As discussed, e.g. in [36], the optimal fusion result can also be represented as a weighted least-squares (WLS) estimate

$$\hat{\underline{x}}^{\mathrm{WLS}} = \arg\min_{\underline{x}} [\underline{\hat{m}} - \mathbf{H}\,\underline{x}]^{\mathrm{T}} \mathbf{J}^{-1} [\underline{\hat{m}} - \mathbf{H}\,\underline{x}]\,, \tag{5}$$

with $\underline{\hat{m}} = \begin{bmatrix} \hat{\underline{x}}^{i} & \hat{\underline{x}}^{j} \end{bmatrix}^{\mathrm{T}}$ and the matrix $\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \end{bmatrix}^{\mathrm{T}}$ which determines how the local states map into the global state vector. The solution to formula (5) is a gain matrix according to

$$\mathbf{K} = \begin{bmatrix} \mathbf{F}^{i} & \mathbf{F}^{j} \end{bmatrix} = \big(\mathbf{H}^{\mathrm{T}} \mathbf{J}^{-1} \mathbf{H}\big)^{-1} \mathbf{H}^{\mathrm{T}} \mathbf{J}^{-1}\,.$$

For two sensor nodes, the fusion gains can be calculated according to the Bar-Shalom/Campo formulas [11] by

$$\mathbf{F}^{j} = \big(\mathbf{P}^{i} - \mathbf{P}^{i,j}\big)\big(\mathbf{P}^{i} + \mathbf{P}^{j} - \mathbf{P}^{i,j} - \mathbf{P}^{j,i}\big)^{-1} \quad \text{and} \quad \mathbf{F}^{i} = \mathbf{I} - \mathbf{F}^{j}\,. \tag{6}$$

Then, the fusion rule can be written as

$$\mathbf{P}^{\mathrm{f}} = \big(\mathbf{H}^{\mathrm{T}} \mathbf{J}^{-1} \mathbf{H}\big)^{-1}, \tag{7}$$

$$\hat{\underline{x}}^{\mathrm{f}} = \mathbf{K}\underline{\hat{m}} = \mathbf{P}^{\mathrm{f}} \mathbf{H}^{\mathrm{T}} \mathbf{J}^{-1} \underline{\hat{m}}\,. \tag{8}$$

Further, from (5) follows that the fusion result is unbiased. This formula can only be solved optimally if the joint covariance matrix $\mathbf{J}$ is available. The entries on the main diagonal are the covariances of the local filters and thus known. The entries on the off-diagonals, on the other hand, are caused by dependent information shared between the individual sensor nodes, and they are usually hard to keep track of.

There are several sources of correlated estimation errors in distributed state estimation problems [13], namely:

1) common prior information,
2) common process noise, and
3) common measurement information.

Common prior information occurs when the local Kalman filters are initialized with the same information, e.g., the same prior state estimate and the same prior covariance matrix. But even with independent initialization of local filters, every sensor node is affected by the same process noise, which leads to correlated estimation errors between state estimates. The local Kalman filters assume conditional independence of measurements, which are then incorporated into the local state estimates. Due to the spread of information throughout the network and further processing, measurement information can be incorporated into several state estimates. This double-counting of sensor data causes additional correlations. Only proper treatment of these correlations allows correct and consistent fusion results.

Optimal fusion is an essential aspect of distributed estimation, and several authors discussed the optimality of the fusion of state estimates, e.g., [36], [37]. However, the fusion of state estimates is not equal to the minimum mean squared error (MMSE) sense in which a central Kalman filter can utilize measurements. Therefore, we want to distinguish between a central Kalman filter and the optimal centralized fusion in this paper.

2.2. Correlations due to Common Process Noise and Common Prior Information

In systems with a central fusion node, see Fig. 1(a), state estimates are correlated due to common process noise and common prior information. When all processing steps are known, the cross-covariances between state estimates can be calculated recursively [11]. During the time update, the process noise is incorporated and the cross-covariance matrix is updated leading to the recursive formula

$$\begin{aligned}
\mathbf{P}^{i,j}_{k|k-1} &= E[(\hat{\underline{x}}^{i}_{k|k-1} - \underline{\boldsymbol{x}}_{k})(\hat{\underline{x}}^{j}_{k|k-1} - \underline{\boldsymbol{x}}_{k})^{\mathrm{T}}] \\
&= E\Big[\big(\mathbf{A}_k \hat{\underline{x}}^{i}_{k-1|k-1} - (\mathbf{A}_k \underline{\boldsymbol{x}}_{k-1} + \boldsymbol{w}_k)\big)\big(\mathbf{A}_k \hat{\underline{x}}^{j}_{k-1|k-1} - (\mathbf{A}_k \underline{\boldsymbol{x}}_{k-1} + \boldsymbol{w}_k)\big)^{\mathrm{T}}\Big] \\
&= \mathbf{A}_k E\Big[\big(\hat{\underline{x}}^{i}_{k-1|k-1} - \underline{\boldsymbol{x}}_{k-1}\big)\big(\hat{\underline{x}}^{j}_{k-1|k-1} - \underline{\boldsymbol{x}}_{k-1}\big)^{\mathrm{T}}\Big] \mathbf{A}_k^{\mathrm{T}} + E\Big[\boldsymbol{w}_k (\boldsymbol{w}_k)^{\mathrm{T}}\Big] \\
&= \mathbf{A}_k \mathbf{P}^{i,j}_{k-1|k-1} \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k\,,
\end{aligned} \tag{9}$$

where $\mathbf{P}^{i,j}_{k-1|k-1}$ for time step $k=1$ is the common prior covariance $\mathbf{P}_{0|0}$. During the measurement update, the cross-covariance is updated using the Kalman filter gain $\mathbf{K}^i_k$ by

$$\begin{aligned}
\mathbf{P}^{i,j}_{k|k} &= E[(\underline{\hat{x}}^i_{k|k} - \underline{\boldsymbol{x}}_k)(\underline{\hat{x}}^j_{k|k} - \underline{\boldsymbol{x}}_k)^{\mathrm{T}}] \\
&= E\Big[\big(\underline{\hat{x}}^i_{k|k-1} + \mathbf{K}^i_k \boldsymbol{z}^i_k - \underline{\boldsymbol{x}}_k\big)\big(\underline{\hat{x}}^j_{k|k-1} + \mathbf{K}^j_k \boldsymbol{z}^j_k - \underline{\boldsymbol{x}}_k\big)^{\mathrm{T}}\Big] \\
&= E\Big[\big(\underline{\hat{x}}^i_{k|k-1} + \mathbf{K}^i_k (\boldsymbol{v}^i_k - \mathbf{C}^i_k \underline{\hat{x}}^i_{k|k-1}) - \underline{\boldsymbol{x}}_k\big)\big(\underline{\hat{x}}^j_{k|k-1} + \mathbf{K}^j_k (\boldsymbol{v}^j_k - \mathbf{C}^j_k \underline{\hat{x}}^j_{k|k-1}) - \underline{\boldsymbol{x}}_k\big)^{\mathrm{T}}\Big] \\
&= \big(\mathbf{I} - \mathbf{K}^i_k \mathbf{C}^i_k\big) E\Big[\big(\underline{\hat{x}}^i_{k|k-1} - \underline{\boldsymbol{x}}_k\big)\big(\underline{\hat{x}}^j_{k|k-1} - \underline{\boldsymbol{x}}_k\big)^{\mathrm{T}}\Big]\big(\mathbf{I} - \mathbf{K}^j_k \mathbf{C}^j_k\big)^{\mathrm{T}} + E\Big[\boldsymbol{v}^i_k (\boldsymbol{v}^j_k)^{\mathrm{T}}\Big] \\
&= \mathbf{L}^i_k \mathbf{P}^{i,j}_{k|k-1} (\mathbf{L}^j_k)^{\mathrm{T}}, \qquad (10)
\end{aligned}$$

where $\mathbf{L}^i_k = \mathbf{I} - \mathbf{K}^i_k \mathbf{C}^i_k$ and $E\Big[\boldsymbol{v}^i_k (\boldsymbol{v}^j_k)^{\mathrm{T}}\Big] = \mathbf{0}$ because the measurement noises are mutually independent. This recursive formulation can also be rewritten explicitly as a sum of the covariances

$$\mathbf{P}^{i,j}_{k|k} = \mathbf{T}^i_{0,k} \mathbf{P}_{0|0} (\mathbf{T}^j_{0,k})^{\mathrm{T}} + \sum_{\tau=1}^{k} \mathbf{T}^i_{\tau,k} \mathbf{Q}_\tau (\mathbf{T}^j_{\tau,k})^{\mathrm{T}}, \qquad (11)$$

where at every time step τ we include the new process noise $\mathbf{Q}_\tau$. The matrix $\mathbf{T}_{\tau,k}$ denote the individual matrix transformations that are a result of the local Kalman filters (see equations (9) and (10)). In large sensor networks, keeping track of these correlations can be cumbersome and often infeasible as it requires full communication of all processing steps. Therefore, the methods in [34], [35] propose the use of square-root decompositions to keep track of correlated estimation errors.

3. Square-Root Decomposition of Common Process Noise

The following section revisits our previous work about the square-root decomposition algorithm. It was originally only formulated for the fusion in centralized sensor networks with only one dedicated fusion center. The basic idea is a sliding window mechanism for a square-root decomposition of the track correlations. Every node updates and saves its history of processing steps in a matrix containing all square-root decompositions of common prior information and common process noise. During the fusion step, every node transmits its state estimate, covariance matrix, and square-root matrix. The square-root matrix allows to reconstruct the joint covariance matrix to fuse the local estimates according to (7) and (8). The recursive formula of (11) is reformulated as a square-root decomposition

$$\begin{aligned}
\mathbf{P}^{i,j}_{k|k} &= \mathbf{T}^i_{0,k} \sqrt{\mathbf{P}_{0|0}} (\sqrt{\mathbf{P}_{0|0}})^{\mathrm{T}} (\mathbf{T}^j_{0,k})^{\mathrm{T}} \\
&\quad + \sum_{\tau=1}^{k} \mathbf{T}^i_{\tau,k} \sqrt{\mathbf{Q}_\tau} (\sqrt{\mathbf{Q}_{\tau,k}})^{\mathrm{T}} (\mathbf{T}^j_{\tau})^{\mathrm{T}} \\
&= \sum_{\tau=0}^{k} \boldsymbol{\Sigma}^i_{\tau,Q} (\boldsymbol{\Sigma}^j_{\tau,Q})^{\mathrm{T}}.
\end{aligned}$$

Each sensor nodes stores its square-root terms in the matrix

$$\boldsymbol{S}^i_{k,Q} = \Big[\boldsymbol{\Sigma}^i_{0,Q}, \boldsymbol{\Sigma}^i_{1,Q}, \ldots, \boldsymbol{\Sigma}^i_{k,Q}\Big],$$

which includes all noise terms until the current time step k and has the dimension $M = N \times D = N \times (N + (k-1) \times W)$. The calculation of this matrix can be done recursively. At time step $k=0$, it is initialized with

$$\boldsymbol{S}^i_{0,Q} = \boldsymbol{\Sigma}^i_{0,Q} = \sqrt{\mathbf{P}_0},$$

and the matrix is then linearly transformed by the time update and a new noise term $\boldsymbol{\Sigma}^i_{k,Q} = \sqrt{\mathbf{Q}_k}$ is included. Furthermore, the matrix is then updated using the gain matrix of the Kalman filter update $\mathbf{L}^i_k = \mathbf{I} - \mathbf{K}^i_k \mathbf{C}^i_k$

$$\boldsymbol{S}^i_{k,Q} = \mathbf{L}^i_k \big[\mathbf{A}^i_k \boldsymbol{S}^i_{k-1,Q} \;,\; \boldsymbol{\Sigma}^i_{k,Q}\big].$$

When the fusion step is reached, the cross-covariance matrix between node i and node j can be reconstructed as

$$\mathbf{P}^{i,j}_{k,Q} = \sum_{m=0}^{k} \boldsymbol{\Sigma}^i_{m,Q} (\boldsymbol{\Sigma}^j_{m,Q})^{\mathrm{T}} = \boldsymbol{S}^i_{k,Q} (\boldsymbol{S}^j_{k,Q})^{\mathrm{T}}. \qquad (12)$$

By including a new process noise term at every time update, the square-root decomposition matrix $\boldsymbol{S}^i_{k,Q}$ will continue to grow linearly in size. Since communication bandwidth is limited in sensor networks, we need to find a trade-off between the optimal decomposition of correlated estimation errors and the communication capacity.

3.1. Limiting the Number of Square-Root Decomposition Terms for Process Noise and Common Prior Information

In order to keep the number of entries in the square-root decomposition matrix constant, the square-root matrix will be decomposed [35] into two parts

$$\boldsymbol{S}_k^i = \left[\boldsymbol{S}_{k,\mathcal{T}_Q}^i \ , \ \boldsymbol{S}_{k,\Omega}^i\right],$$

where $\boldsymbol{S}_{k,\mathcal{T}_Q}^i$ is a moving horizon square-root decomposition matrix

$$\boldsymbol{S}_{k,\mathcal{T}_Q}^i = \left[\boldsymbol{\Sigma}_{k-\mathcal{T}_Q+1}^i, \boldsymbol{\Sigma}_{k-\mathcal{T}_Q+2}^i, \ldots, \boldsymbol{\Sigma}_k^i\right] \tag{13}$$

that will only include the dependent noise terms up to a user-defined time horizon $\mathcal{T}_Q$. The remaining noise terms will be removed from the square-root matrix and summarized in a residual $\boldsymbol{S}_{k,\Omega}^i$. This residual has to be bounded in order to obtain a consistent fusion result. To formulate the fusion rule, we consider the optimal joint covariance matrix

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{P}_k^i & \mathbf{P}_k^{i,j} \\ \mathbf{P}_k^{j,i} & \mathbf{P}_k^j \end{bmatrix}.$$

We can now decompose this matrix into a part $\mathbf{P}_{k,\mathcal{T}_Q}^{i,j}$ that we can reconstruct and a part $\mathbf{P}_{k,\Omega}^{i,j}$ that is correlated but whose exact correlation we cannot reconstruct anymore, i.e.,

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{P}_k^i & \mathbf{P}_{k,\mathcal{T}_Q}^{i,j} + \mathbf{P}_{k,\Omega}^{i,j} \\ \mathbf{P}_{k,\mathcal{T}_Q}^{j,i} + \mathbf{P}_{k,\Omega}^{j,i} & \mathbf{P}_k^j \end{bmatrix}.$$

This residual can be calculated recursively and includes all correlated noise terms not included in the square-root matrix $\boldsymbol{S}_{k,\mathcal{T}}^i$. With the residual, we obtain

$$\boldsymbol{S}_{k,\Omega}^i \left(\boldsymbol{S}_{k,\Omega}^i\right)^{\mathrm{T}} = \boldsymbol{\Omega}_{k,Q}^i \,. \tag{14}$$

We now aim to find a bound according to

$$\begin{bmatrix} \frac{1}{\omega}\boldsymbol{\Omega}_{k,Q}^i & 0 \\ 0 & \frac{1}{1-\omega}\boldsymbol{\Omega}_{k,Q}^j \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{\Omega}_{k,Q}^i & \mathbf{P}_{k,\Omega}^{i,j} \\ \mathbf{P}_{k,\Omega}^{j,i} & \boldsymbol{\Omega}_{k,Q}^j \end{bmatrix}.$$

Finally, we can now formulate the new suboptimal joint covariance matrix

$$\widetilde{\mathbf{J}}_k = \begin{bmatrix} \mathbf{P}_k^i - \boldsymbol{\Omega}_{k,Q}^i & \mathbf{P}_{k,\mathcal{T}_Q}^{i,j} \\ \mathbf{P}_{k,\mathcal{T}_Q}^{j,i} & \mathbf{P}_k^j - \boldsymbol{\Omega}_{k,Q}^j \end{bmatrix} + \begin{bmatrix} \frac{1}{\omega}\boldsymbol{\Omega}_{k,Q}^i & 0 \\ 0 & \frac{1}{1-\omega}\boldsymbol{\Omega}_{k,Q}^j \end{bmatrix} \geq \mathbf{J}_k, \tag{15}$$

which we will use for the fusion step according to formulas (3) and (4). The weighting factors ω can be found by minimizing the fused covariance matrix according to formula (7). Alternatively, an approximate solution such as the one proposed by [34], [38] can be used. Although suboptimal, we used the latter approach for its simple implementation and fast execution time. The weighting factor can be calculated by

$$\omega = \frac{1/\mathrm{tr}(\boldsymbol{\Omega}_Q^i)}{1/\mathrm{tr}(\boldsymbol{\Omega}_Q^i) + 1/\mathrm{tr}(\boldsymbol{\Omega}_Q^j)}.$$

Afterwards the formula given in (6) yields

$$\mathbf{F}^j = \left(\mathbf{P}^i + \frac{1}{\omega}\boldsymbol{\Omega}_Q^i - \mathbf{P}_{\mathcal{T}_Q}^{i,j}\right)\left(\mathbf{P}^i + \frac{1}{\omega}\boldsymbol{\Omega}_Q^i + \mathbf{P}^j + \frac{1}{1-\omega}\boldsymbol{\Omega}_Q^j - \mathbf{P}_{\mathcal{T}_Q}^{i,j} - \mathbf{P}_{\mathcal{T}_Q}^{j,i}\right)^{-1}.$$

Last, the fused covariance and fused state can be calculated according to equations (7) and (8).

4. Extension to the Fusion in Decentralized Sensor Networks

The square-root decomposition enables the nodes to encode correlated process noise and correlated prior information in a distributed fashion. The central node in Fig. 1(a) does not need to keep track of the correlations, processing steps, or number of nodes as all the required information is provided by the nodes themselves. Modifications to the square-root decomposition are necessary when nodes are organized in hierarchical network topologies as shown in Fig. 1(b), where intermediate fusion nodes exist. Each fusion step alters the correlation structure among the nodes, which has to be encoded properly and is discussed in Sec. 4.1. The decentralized network architecture depicted in Fig. 1(c) exhibits cycles that lead to double counting of information. Sec. 4.2 discusses how additional data structures can be introduced to cover correlations due to double-counting of measurements and thus correlated measurement errors.

4.1. Hierarchical Fusion

In a hierarchical fusion architecture, nodes may fuse estimates and pass them to the upper layer for a subsequent fusion step. Hence, such intermediate fusion nodes have to take into account correlations for the fusion but simultaneously have to compute an updated square-root decomposition for the subsequent fusion steps. Each node i can fuse its estimate with an estimate received from node j by using the fusion formulas (3) and (4). The required cross-covariance matrices $\mathbf{P}^{i,j} = \left(\mathbf{P}^{j,i}\right)^{\mathrm{T}}$ are obtained by the square-root decomposition, i.e., by using (12).

For the subsequent fusion layer, the square-root decomposition needs to encode the correlation structure of the fusion result $\hat{\underline{x}}^{\mathrm{f}}$. The cross-covariance matrix for this fusion result $\hat{\underline{x}}^{\mathrm{f}}$ and the estimate $\hat{\underline{x}}^{l}$ of a third node l yields

$$\begin{aligned}\mathbf{P}^{\mathrm{f},l} &= E[(\hat{\underline{x}}^{\mathrm{f}} - \underline{x})(\hat{\underline{x}}^{l} - \underline{x})^{\mathrm{T}}] \\ &= E[(\mathbf{F}^{i}\,\hat{\underline{x}}^{i} + \mathbf{F}^{j}\,\hat{\underline{x}}^{j} - \underline{x})(\hat{\underline{x}}^{l} - \underline{x})^{\mathrm{T}}] \\ &= \mathbf{F}^{i}\,\mathbf{P}^{i,l} + \mathbf{F}^{j}\,\mathbf{P}^{j,l}\ .\end{aligned}$$

The dependencies $\mathbf{P}^{i,l}$ and $\mathbf{P}^{j,l}$ are given by the corresponding square-root decompositions, i.e.,

$$\mathbf{P}^{i,l} = \boldsymbol{S}_{Q}^{i}\big(\boldsymbol{S}_{Q}^{l}\big)^{\mathrm{T}} \text{ and } \mathbf{P}^{j,l} = \boldsymbol{S}_{Q}^{j}\big(\boldsymbol{S}_{Q}^{l}\big)^{\mathrm{T}}\ .$$

Hence, the fused square-root decomposition for the $\mathbf{P}^{\mathrm{f},l}$ has the form

$$\boldsymbol{S}_{Q}^{\mathrm{f}} = \mathbf{F}^{i}\boldsymbol{S}_{Q}^{i} + \mathbf{F}^{j}\boldsymbol{S}_{Q}^{j}\,, \tag{16}$$

which gives $\mathbf{P}^{\mathrm{f},l} = \boldsymbol{S}_{Q}^{\mathrm{f}}(\boldsymbol{S}_{Q}^{l})^{\mathrm{T}}$ for any l.

For a finite horizon $\mathcal{T}_Q$, $\boldsymbol{S}^{\mathrm{f}}$ only partially covers the correlations, and the fusion node also has to update the residual term (14). According to the chosen weight ω in (15), the residual becomes

$$\begin{aligned}\boldsymbol{\Omega}_{Q}^{\mathrm{f}} &= \tfrac{1}{\omega}\mathbf{F}^{i}\boldsymbol{\Omega}_{Q}^{i}\big(\mathbf{F}^{i}\big)^{\mathrm{T}} + \tfrac{1}{1-\omega}\mathbf{F}^{j}\boldsymbol{\Omega}_{Q}^{j}\big(\mathbf{F}^{j}\big)^{\mathrm{T}} \\ &\geq \mathbf{F}^{i}\boldsymbol{\Omega}_{Q}^{i}\big(\mathbf{F}^{i}\big)^{\mathrm{T}} + \mathbf{F}^{i}\boldsymbol{\Omega}_{Q}^{i,j}\big(\mathbf{F}^{j}\big)^{\mathrm{T}} \\ &\quad + \mathbf{F}^{j}\boldsymbol{\Omega}_{Q}^{j,i}\big(\mathbf{F}^{i}\big)^{\mathrm{T}} + \mathbf{F}^{j}\boldsymbol{\Omega}_{Q}^{j}\big(\mathbf{F}^{j}\big)^{\mathrm{T}}\ ,\end{aligned} \tag{17}$$

which is a bound since any information about $\boldsymbol{\Omega}_{Q}^{i,j}$ has been discarded.

4.2. Double Counting

Double counting occurs when two nodes i and j fuse their estimates for a second time. In other words, the information sent out by node i circles back to this node over possibly multiple hops and processing steps. Not only common process noise then leads to correlations, but also measurements incorporated in the estimates reappear at the nodes and introduce additional correlations. In the latter case, two estimates are to be fused that share the same information. The cross-covariance matrix between the fused estimate $\hat{\underline{x}}^{\mathrm{f}}$ and the estimate $\hat{\underline{x}}^{i}$ of node i yields

$$\begin{aligned}\mathbf{P}^{\mathrm{f},i} &= E[(\hat{\underline{x}}^{\mathrm{f}} - \underline{x})(\hat{\underline{x}}^{i} - \underline{x})^{\mathrm{T}}] \\ &= E[(\mathbf{F}^{i}\,\hat{\underline{x}}^{i} + \mathbf{F}^{j}\,\hat{\underline{x}}^{j} - \underline{x})(\hat{\underline{x}}^{i} - \underline{x})^{\mathrm{T}}] \\ &= \mathbf{F}^{i}\,\mathbf{P}^{i,i} + \mathbf{F}^{j}\,\mathbf{P}^{j,i}\,.\end{aligned}$$

The cross-covariance $\mathbf{P}^{j,i}$ can be calculated as discussed in section 2.2. The matrix $\mathbf{P}^{i,i}$ represents the correlated estimation errors of sensor node i and is equal to the covariance matrix

$$\begin{aligned}\mathbf{P}_{k|k}^{i} &= E[(\hat{\underline{x}}_{k|k}^{i} - \underline{\boldsymbol{x}}_{k})(\hat{\underline{x}}_{k|k}^{i} - \underline{\boldsymbol{x}}_{k})^{\mathrm{T}}] \\ &= E\Big[\big(\hat{\underline{x}}_{k|k-1}^{i} + \mathbf{K}_{k}^{i}\boldsymbol{z}_{k}^{i} - \underline{\boldsymbol{x}}_{k}\big)\big(\hat{\underline{x}}_{k|k-1}^{i} + \mathbf{K}_{k}^{i}\boldsymbol{z}_{k}^{i} - \underline{\boldsymbol{x}}_{k}\big)^{\mathrm{T}}\Big] \\ &= E\Big[\big(\hat{\underline{x}}_{k|k-1}^{i} + \mathbf{K}_{k}^{i}(\boldsymbol{v}_{k}^{i} - \mathbf{C}_{k}^{i}\hat{\underline{x}}_{k|k-1}^{i}) - \underline{\boldsymbol{x}}_{k}\big)\big(\hat{\underline{x}}_{k|k-1}^{i} + \mathbf{K}_{k}^{i}(\boldsymbol{v}_{k}^{i} - \mathbf{C}_{k}^{j}\hat{\underline{x}}_{k|k-1}^{i}) - \underline{\boldsymbol{x}}_{k}\big)^{\mathrm{T}}\Big] \\ &= \mathbf{L}_{k}^{i}\mathbf{P}_{k|k-1}^{i}(\mathbf{L}_{k}^{i})^{\mathrm{T}} + \mathbf{K}_{k}^{i}\mathbf{R}^{i}\big(\mathbf{K}_{k}^{i}\big)^{\mathrm{T}}\,,\end{aligned}$$

with the Kalman filter update $\mathbf{L}_{k}^{i} = \mathbf{I} - \mathbf{K}_{k}^{i}\mathbf{C}_{k}^{i}$. For this reason, each node i needs to keep track of an additional list of measurement noise terms

$$\boldsymbol{S}_{k,R^{i}}^{i} = \Big[\boldsymbol{\Sigma}_{0,R^{i}}^{i}, \boldsymbol{\Sigma}_{1,R^{i}}^{i}, \ldots, \boldsymbol{\Sigma}_{k,R^{i}}^{i}\Big]\,, \tag{18}$$

to account for double counting of measurements. It is initialized at time step $k = 1$ with

$$\boldsymbol{S}_{1,R^{i}}^{i} = \boldsymbol{\Sigma}_{1,R^{i}}^{i} = \mathbf{K}_{1}^{i}\sqrt{\mathbf{R}_{1}^{i}}\,,$$

where $\mathbf{R}_1^i$ is the measurement covariance matrix of the first measurement (2) acquired by node i. The matrix $\mathbf{K}_1^i$ is the Kalman gain used in this measurement update. The matrix $\boldsymbol{S}_{k,R^i}^i$ is recursively updated according to[1]

$$\boldsymbol{S}_{k,R^i}^i = \left[\mathbf{L}_k^i \mathbf{A}_k^i \boldsymbol{S}_{k-1,R^i}^i \;,\; \boldsymbol{\Sigma}_{k,R^i}^i\right] \tag{19}$$

with

$$\boldsymbol{\Sigma}_{k,R^i}^i = \mathbf{K}_k^i \sqrt{\mathbf{R}_k^i}\,.$$

When two sensor nodes exchange estimates for fusion, they also pass on all the square-root matrices. These matrices need to be kept separate from each other to trace back possible sources of double counting. Node i that receives an estimate from node j then also keeps and manages the set $\boldsymbol{S}_{k,R^j}^i$, which is the corresponding set (19) from node j. The own and the received square-root matrices are updated similarly to (16) and (17) by

$$\begin{aligned}\boldsymbol{S}_{R^i}^{\mathrm{f}} &= \mathbf{F}^i \boldsymbol{S}_{R^i}^i + \mathbf{F}^j \boldsymbol{S}_{R^i}^j\,,\\ \boldsymbol{S}_{R^j}^{\mathrm{f}} &= \mathbf{F}^i \boldsymbol{S}_{R^j}^i + \mathbf{F}^j \boldsymbol{S}_{R^j}^j\,.\end{aligned}$$

Bookkeeping of the received $\boldsymbol{S}_{k,R^j}^i$ resembles (19). However, it differs in that it is filled with zeros during further processing according to

$$\boldsymbol{S}_{k,R^j}^i = \mathbf{L}_k^i \left[\mathbf{A}_k^i \boldsymbol{S}_{k-1,R^j}^{i,j} \;,\; \mathbf{0}\right] \tag{20}$$

as the measurement noise affecting node j is uncorrelated with the estimation errors at node i for the following time steps.

The square-root matrix $\boldsymbol{S}_{k,R^i}^i$ can be used in a later fusion step to retrieve the cross-covariances stemming from the previous fusion step by

$$\mathbf{P}_{k,R}^{i,j} = \boldsymbol{S}_{k,R^i}^i (\boldsymbol{S}_{k,R^i}^j)^{\mathrm{T}} + \boldsymbol{S}_{k,R^j}^i (\boldsymbol{S}_{k,R^j}^j)^{\mathrm{T}}\,, \tag{21}$$

where $\boldsymbol{S}_{k,R^i}^j$ is the common information with node i that has been tracked in node j. More precisely, $\boldsymbol{S}_{k,R^i}^j$ is the corresponding set to (20) that was generated by node j when it received information from i. The reconstructed cross-covariance matrix (21) has to be combined with $\mathbf{P}_{k,Q}^{i,j}$ representing the common process noise, which finally results in the full cross-covariance matrix

$$\mathbf{P}_k^{i,j} = \mathbf{P}_{k,Q}^{i,j} + \mathbf{P}_{k,R}^{i,j}\,.$$

The amount of data that need to be stored and updated by each node grows linearly over time. Especially in networks with many sensor nodes, conservative bounding techniques can allow the nodes to surpass this burden.

4.2.1. Limiting the Number of Square-Root Decomposition Terms for Measurement Noise: Following the concept in Sec. 3.1, we limit the number of processing steps encoded in the square-root decompositions to a fixed time horizon $\mathcal{T}_R$. The matrix (18) becomes

$$\boldsymbol{S}_R^i = \left[\boldsymbol{\Sigma}_{R,k-\mathcal{T}_R+1}^i, \boldsymbol{\Sigma}_{R,k-\mathcal{T}_R+2}^i, \ldots, \boldsymbol{\Sigma}_{R,k}^i\right],$$

which has a constant number of entries. The remainder of the matrix is summarized in the residual term $\boldsymbol{\Omega}_R^i$. When two estimates are fused, a bound on the residual matrix as in (17) has to be computed by

$$\boldsymbol{\Omega}_R^{\mathrm{f}} = \tfrac{1}{\omega}\mathbf{F}^i \boldsymbol{\Omega}_R^i \left(\mathbf{F}^i\right)^{\mathrm{T}} + \tfrac{1}{1-\omega}\mathbf{F}^j \boldsymbol{\Omega}_R^j \left(\mathbf{F}^j\right)^{\mathrm{T}}.$$

This bound also has to be combined with the residual bound (17) for the process noise.

4.2.2. Keeping Track of Uncorrelated Measurements: The treatment of correlated measurement information and double counting can be simplified by computing a more general bound on the measurement covariance. This approach circumvents the explicit bookkeeping (18) of the information shared through the fusion of estimates.

The local covariance matrix of sensor node i is rewritten as

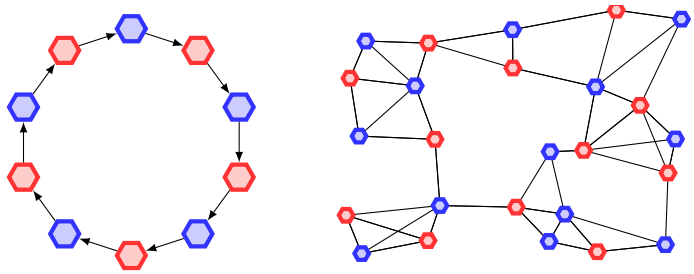
$$\mathbf{P}^i = \mathbf{P}_{Q,\mathcal{T}_Q} + \mathbf{P}_{Q,\Omega} + \mathbf{P}_R\,,$$

where $\mathbf{P}_{Q,\mathcal{T}_Q}$ accounts for the reconstructable cross-covariance matrix using (13), $\mathbf{P}_{Q,\Omega}$ accounts for the residual (14), and $\mathbf{P}_R$ represents possibly correlated measurement noise. We further separate this into

$$\mathbf{P}_R = \mathbf{P}_R^+ + \mathbf{P}_R^-\,,$$

where $\mathbf{P}_R^+$ denotes correlated measurement noise and $\mathbf{P}_R^-$ uncorrelated measurement noise. We can safely assume that measurements that have been obtained between fusion steps and thus have not been shared are uncorrelated. Therefore, only

[1] Note that $\mathbf{L}_k^i$ in [1] should be inside the brackets.



(a) Ring Topology (b) Net Topology

Fig. 2: Network topologies. Magenta nodes are using one measurement model and blue nodes the other measurement model.

the part accounting for information that has been shared with other sensor nodes before is correlated and needs to be bounded. The uncorrelated measurement noise residual $\mathbf{P}_{k,R}^{-}$ can be calculated recursively

$$\mathbf{P}_{k,R}^{-} = \mathbf{L}\mathbf{A}\mathbf{P}_{k-1,R}^{-}\mathbf{A}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}} + \mathbf{K}_k^i\mathbf{R}_1^i\left(\mathbf{K}_k^i\right)^{\mathrm{T}}.$$

To ensure the correctness of this assumption, $\mathbf{P}_{k,R}^{-}$ will be reset to the zero matrix when the fusion step has been executed or the information has been shared with other sensor nodes. The correlated measurement residual is calculated by

$$\mathbf{\Omega}_R^i = \mathbf{P}^i - \boldsymbol{S}_Q^i\left(\boldsymbol{S}_Q^i\right)^{\mathrm{T}} - \mathbf{\Omega}_Q^i - \mathbf{P}_R^{-}.$$

The bounded part of the joint covariance matrix becomes

$$\mathbf{\Omega}_k^i = \mathbf{\Omega}_{k,Q}^i + \mathbf{\Omega}_{k,R}^i.$$

The rest of the fusion step is analogous to (15).

5. Evaluation

The following section features three distinct examples to highlight the performance of the proposed algorithm under different conditions. First, we discuss an example using only two sensor nodes that constantly exchange information, which leads to highly correlated estimates. Second, we discuss the convergence rate of the proposed algorithm and compare it with standard consensus algorithms. Last, a tracking example using 25 heterogeneous sensor nodes in a sparse network but with synchronized fusion steps is analyzed.

5.1. Two Sensor Nodes

We consider two sensor nodes A and B, which observe the discrete-time time-invariant linear stochastic system in (1) with the parameters

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}, \ \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \Delta T = 0.1.$$

Both sensor nodes draw observations using the linear measurement model (2), where every measurement is corrupted by additive-white Gaussian noise $\underline{v}_k^i$ with covariance matrix $\mathbf{R}^A = \mathbf{R}^B = 50$ and measurement matrices

$$\mathbf{C}^A = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \mathbf{C}^B = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

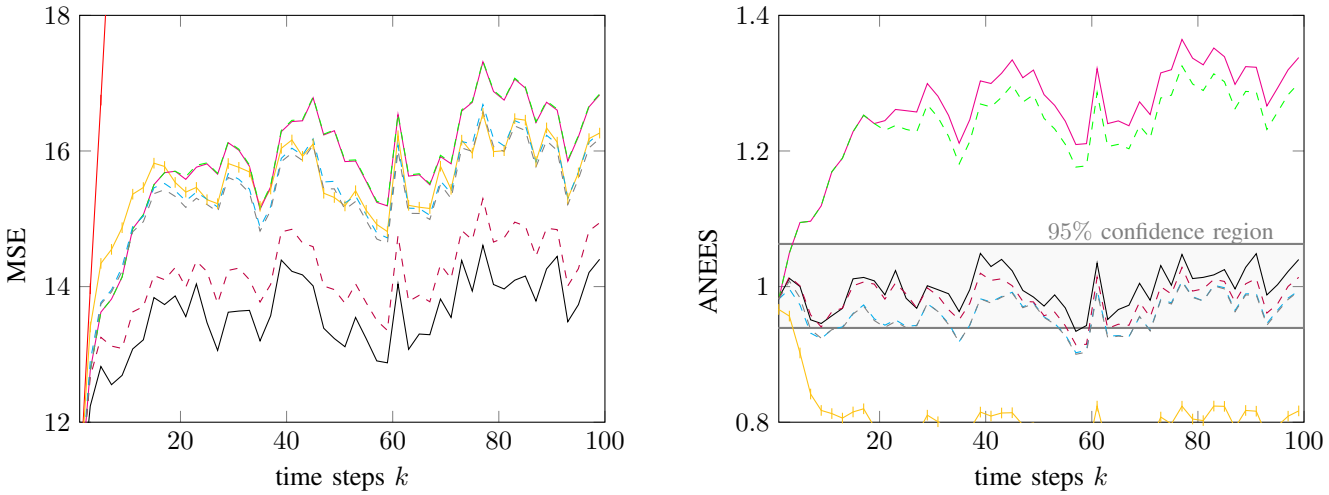
Both sensor nodes are initialized with $\mathbf{P}_0 = 5\mathbf{Q}$ and $\underline{\hat{x}}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$. The data exchange between the two nodes is performed as follows

1) both sensor nodes execute a local filter update,
2) node A sends its local information to node B,
3) node B fuses information according to the selected fusion method and reinitializes its local state and covariance matrix with new fused information,
4) both sensor nodes execute a local filter update,
5) node B sends its local information to node A,
6) node A fuses information according to the selected fusion method and reinitializes its local state and covariance matrix with new fused information,
7) repeat from beginning.

We calculate the mean squared error (MSE) of both sensor nodes and then calculate the average. Fig. 3(a) shows the averaged MSE of both sensor nodes for 1000 Monte Carlo Runs (MCR). The results are compared with the optimal fusion result.

Table I: Abbreviations and parameterizations for two sensor nodes example

Method	Short		Parameterization
Covariance intersection [16]	CI		-
Naïve fusion [15]	Naïve		-
Optimal fusion [11] (central)	Optimal		-
Square-root decomp. (Sec. 3)	SqDF_{no}		$\mathcal{T}_Q = 5$
Square-root decomp. (Sec. 3.1)	SqDF_{Qb}		$\mathcal{T}_Q = 5$
Square-root decomp. (Sec. 4.22)	SqDF_{Rbp}		$\mathcal{T}_Q = 5$
Square-root decomp. (Sec. 4.21)	SqDF_{Rb1}		$\mathcal{T}_Q = 5$, $\mathcal{T}_R = 5$
Square-root decomp. (Sec. 4.21)	SqDF_{Rb2}		$\mathcal{T}_Q = 5$, $\mathcal{T}_R = 2$



(a) Averaged MSE of both sensor nodes.

(b) Averaged ANEES of both sensor nodes.

Fig. 3: Comparison of the fusion results of different algorithms for 1000 Monte Carlo Runs.

The optimal fusion result is obtained by optimally keeping track of the cross-covariance matrices between the state estimates and performing the fusion step in one dedicated fusion center using a centralized network topology. After the fusion step is executed, the local state estimates and covariances matrices are reinitialized with the fusion result. This approach shows the lowest MSE as expected. The MSE of the naïve fusion, which neglects the correlations between state estimates, immediately diverges. The proposed square-root decomposition (SqDF) is shown in several configurations. The time horizon for the square-root matrix is $\mathcal{T}_Q = 5$. The square-root decomposition without bounding (SqDF_{no}) shows a relatively high MSE as it does not account for older process noise or any correlation due to measurement noise. Bounding of process noise (SqDF_{Qb}) performs a bit better in comparison as it does bound the process noise but also does not account for possibly correlated measurements. Covariance intersection (CI) performs better than SqDF_{no} and SqDF_{Qb}, but its performance is limited as it cannot account for uncorrelated parts. Using the proposed algorithm with partial bounding of measurement noise (SqDF_{Rbp}, see Sec. 4.22) shows better performance than covariance intersection, as it can find a tighter bound. The proposed method from Sec. 4.21 using the limited time horizon $\mathcal{T}_R$ for the track-keeping of correlated measurement noise is also compared to the other methods. The square-root decomposition using a time horizon of $\mathcal{T}_R = 5$ (SqDF_{Rb1}) shows the lower mean squared error comparing to all other methods. The square-root decomposition using a smaller time horizon of $\mathcal{T}_R = 2$ (SqDF_{Rb2}) is comparable to the performance of CI.

Fig. 3(b) shows the averaged normalized estimation error squared (ANEES) over both sensor nodes. The ANEES is a measure to determine whether the actual uncertainty matches the expected uncertainty [39]. An ANEES below one indicates a conservative fusion estimate, while an ANEES above one indicates an underestimation of the actual uncertainty. Naïve fusion diverges again very fast and is therefore not included in the plot, and covariance intersection is overly conservative. Both methods without bounding (SqDF_{no} and SqDF_{Qb}) are inconsistent as it would be expected. The algorithm with partial bounding is close to one, meaning that the actual MSE of the fused results matches the covariance matrix. The proposed methods using a limited time horizon to keep track of correlated measurement noise (SqDF_{Rb1} and SqDF_{Rb2}) are very close to the optimal fusion result but slightly more conservative, where SqDF_{Rb2} shows similar performance to the proposed method with the partial bounding of correlated measurement errors (SqDF_{Rbp}).

A summary of all used abbreviations and parameterizations of the used methods can be found in Table I.

Table II: Abbreviations and parameterizations for consensus example

Method	Short		Parameterization
Covariance intersection [16]	CI		-
Naïve fusion [15]	Naïve		-
Optimal fusion [11] (central)	Optimal		-
Square-root decomp. (Sec. 4.22)	SqDF_{Rbp}		$\mathcal{T}_Q = 11$
Square-root decomp. (Sec. 4.2)	SqDF_{Opt}		$\mathcal{T}_Q = 11$, $\mathcal{T}_R = 10$
Square-root decomp. (Sec. 4.21)	SqDF_{Rb1}		$\mathcal{T}_Q = 11$, $\mathcal{T}_R = 5$
Square-root decomp. (Sec. 4.21)	SqDF_{Rb2}		$\mathcal{T}_Q = 11$, $\mathcal{T}_R = 2$
Consensus on Measurements [24]	Cons_{M}		Metropolis weights
Consensus on Information [26]	Cons_{M}		Metropolis weights
Hybrid consensus filter [28]	DHIWCF		Metropolis weights

5.2. Consensus between States

In the following example, we discuss how fast the proposed algorithm convergences towards a global consensus. Consensus problems have been intensively studied in many different contexts [24]. Instead of accounting for dependencies within the network, consensus algorithms average the information of neighboring nodes iteratively until all sensor nodes converged asymptotically to a global estimate [29]. While finding a consensus is usually not the goal of fusion algorithms, it is an interesting problem to investigate the effect of double-counting in sensor networks. This section demonstrates that the careful consideration of dependencies improves the convergence rate towards a global consensus. We define the averaged consensus estimate error (ACEE) that indicates the degree of consensus among estimates from all nodes in the network (see also [28])

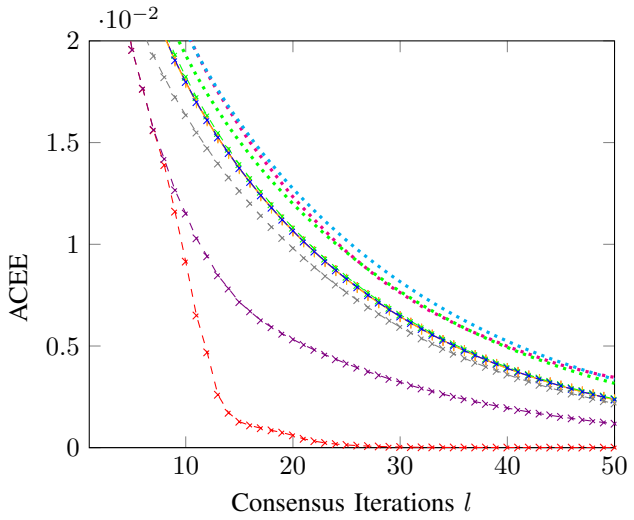
$$\text{ACEE} = \frac{1}{N_s}\sum_{i=1}^{N_s}\left(\hat{\underline{x}}^i - \bar{\underline{x}}\right), \; \bar{\underline{x}} = \frac{1}{N_s}\sum_{i=1}^{N_s}\hat{\underline{x}}^i .$$

We consider a network of ten sensor nodes with ring topology (see Fig. 2(a)). The system description is similar to the one in Example 1, but the measurement covariances are reduced to $\mathbf{R}^A = \mathbf{R}^B = 0.2$ and to decrease oscillation. The sensor nodes alternate between the measurement model of node A and node B, which can also be seen in the figure. The sensor nodes first perform ten filtering steps independently and then communicate their local information towards their neighbors multiple times. The fusion algorithms are also compared with consensus algorithms, specifically consensus on measurements [24] (Cons_{M}), consensus on information [26] (Cons_{I}) and a hybrid consensus method called DHIWCF [28], which performs a consensus on measurement on the first iteration and a consensus on information afterwards. All consensus methods are performed using Metropolis weights. We would like to point out that many consensus algorithms have been proposed in recent years and that the utilized algorithms may not be best tailored to the considered problem. A summary of all used abbreviations and parameterizations of the used methods can be found in Table II.

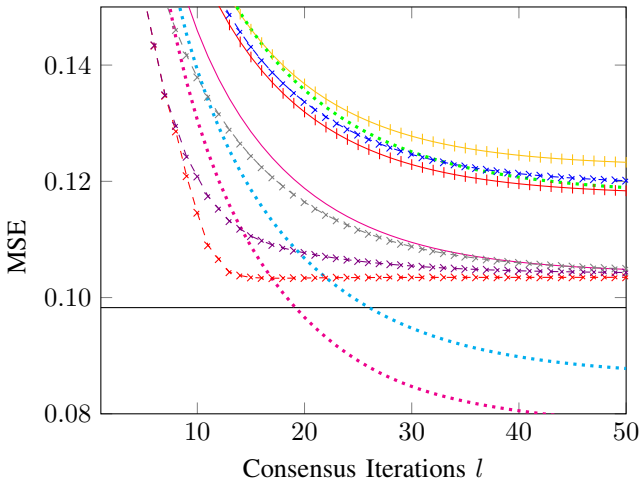
Fig. 4(a) shows the convergence rate of the state estimates. Covariance intersection and naïve fusion show very similar convergence rates. All consensus methods converge only slightly slower. The hybrid consensus algorithm DHIWCF lies between consensus on measurements and consensus on information. Furthermore, we see that the square-root decomposition of the measurement noise improves the convergence rate. Keeping track of all measurements (SqDF_{Opt}) leads to the fastest convergence, followed by the square-root decomposition with a time horizon $\mathcal{T}_R = 3$ (SqDF_{Rb1}) and using a time horizon $\mathcal{T}_R = 1$ (SqDF_{Rb2}). Therefore showing that even a short time horizon for the measurement noise might make a huge difference. The time horizon of the square root matrix keeping track of the process noise is $\mathcal{T}_Q = 11$. Therefore, process noise and common prior information are fully tracked.

For further comparison, we computed the mean squared error (MSE) for all sensor nodes and showed the averaged MSE in Fig. 4(b). Compared with all other fusion methods, the optimal track keeping of correlations achieves the lowest MSE fastest and almost approaches the centralized optimal fusion result. The square-root decomposition with a smaller time horizon SqDF_{Rb1} and SqDF_{Rb2} also performs well but converge more slowly. Consensus on information does not show any performance improvements in comparison to the other fusion methods. On the other hand, consensus on measurements converges slightly slower but outperforms all other methods after 18 time steps. The hybrid method DHIWCF shows slightly lower performance. Both consensus methods reach a lower average mean squared error because the utilization of measurement information is more effective than the exclusive fusion of state estimates.

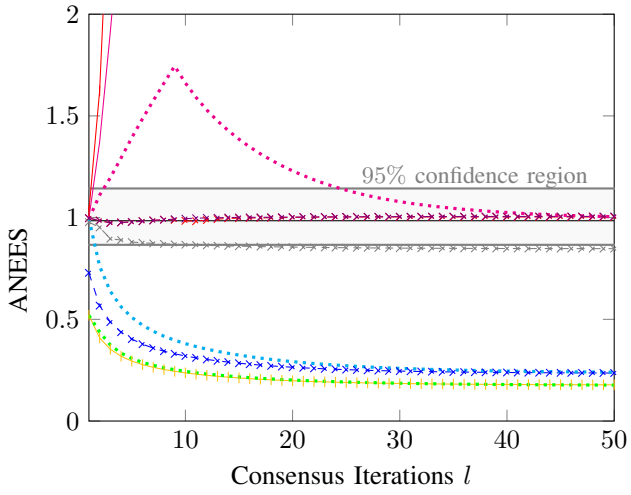
Lastly, in Fig. 4(c), it can be seen that the average ANEES over all sensor nodes in the network is close to the optimal fusion result for SqDF_{Rb1}, SqDF_{Opt}, and SqDF_{Rb2}. All square-root decomposition-based algorithms that bound the measurement partially or fully are very close to the performance of covariance intersection and, therefore, overly conservative. Consensus on information shows similar performance as covariance intersection but performs slightly worse because Metropolis weights do not minimize the trace or determinant of the fused covariance matrix. The performance of consensus on measurements depends on the utilized correction weights to mitigate the averaging of measurements [40]. We chose the correction weight as 2 in the first consensus step when only two measurements are available to the sensor node. Then we increment the correction



(a) Convergence of state estimates to consensus



(b) Averaged MSE over all sensor nodes



(c) Averaged ANEES over all sensor nodes

Fig. 4: Convergence of state estimates towards a common consensus and mean squared error for ring topology with 200 MCR.

weight by one in every consensus step until 10 to account for the ten measurements once a consensus is reached. Because of the averaging characteristics, the ANEES will start to rise as some measurements have higher weights than others during the averaging, leading to double counting. Once the consensus is approached, the ANEES will converge towards 1 again, meaning that the method will be consistent after a certain amount of time. DHIWCF shows slightly less conservative results than covariance intersection. This means that it can reach a relatively low mean squared error while still achieving consistent results, which is an interesting finding. Yet, the best trade-off between convergence rate, mean squared error, and consistency can be achieved using the proposed method.

5.3. Large-Scale Sparse Network

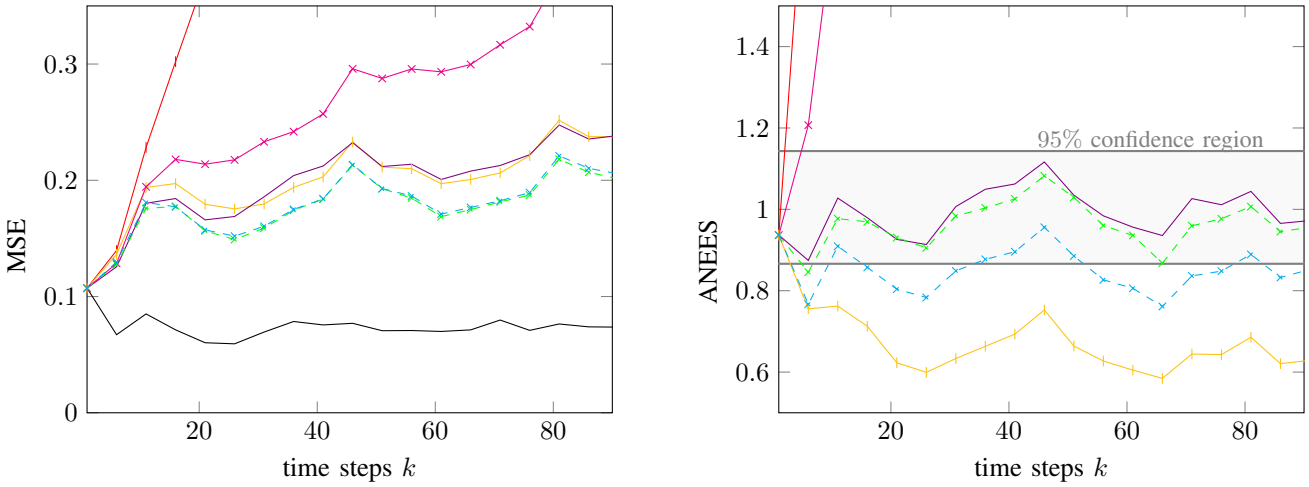
In our last example, we consider a simple tracking example featuring 25 sensor nodes in a sparse network as depicted in Fig. 2(b). Nodes always receive information from the three closest sensor nodes. The movement of the tracked object is described by

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}\underline{\boldsymbol{x}} + \underline{\boldsymbol{w}}_k \text{ with } \underline{\boldsymbol{w}}_k \sim \mathcal{N}\left(\underline{0}, \mathbf{Q}\right),$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = 0.1 \begin{bmatrix} \frac{1}{3}\Delta T & 0 & \frac{1}{2}\Delta T & 0 \\ 0 & \frac{1}{3}\Delta T & 0 & \frac{1}{2}\Delta T \\ \frac{1}{2}\Delta T & 0 & \Delta T & 0 \\ 0 & \frac{1}{2}\Delta T & 0 & \Delta T \end{bmatrix}, \quad \Delta T = 0.1\,.$$

Table III: Abbreviations and parameterizations for large scale network example

Method	Short		Parameterization
Covariance intersection [16]	CI		-
Naïve fusion [15]	Naïve		-
Optimal fusion [11] (central)	Optimal		-
Square-root decomp. (Sec. 3)	SqDF_{no}		$\mathcal{T}_Q = 5$
Square-root decomp. (Sec. 3.1)	SqDF_{Qb}		$\mathcal{T}_Q = 5$
Square-root decomp. (Sec. 4.22)	SqDF_{Rbp}		$\mathcal{T}_Q = 5$
Channel filter [21]	ChF		



(a) Averaged MSE of all 25 sensor nodes.

(b) Averaged ANEES of all 25 sensor nodes.

Fig. 5: Comparison of the fusion results of different algorithms for 100 Monte Carlo Runs.

Referring again to Fig. 2(b), the blue nodes observe the bearing towards a moving target and the red nodes the range. Their observation is described by a nonlinear measurement function

$$\underline{y}_k^i = h^i(\underline{\boldsymbol{x}}_k) + \underline{\boldsymbol{v}}_k \,,$$

where nodes alternate between measuring the bearing or the range towards a moving target

$$h^i(\underline{\boldsymbol{x}}_k) = \begin{cases} \text{atan2}\left(x_{y,k} - P_y^i \,,\, x_{x,k} - P_x^i\right) & \text{if } i \text{ is odd,} \\ \sqrt{\left(x_{x,k} - P_x^i\right)^2 + \left(x_{y,k} - P_y^i\right)^2} & \text{if } i \text{ is even,} \end{cases}$$

with measurement noise

$$\mathbf{R}^i = \left(\tfrac{2\pi}{180}\right)^2 \text{ if } i \text{ is odd, or } \mathbf{R}^i = 0.01\,\text{m}^2 \text{ if } i \text{ is even}$$

at the sensor nodes position $P^i = [P_x^i, P_y^i]^{\text{T}}$.

The nodes are placed at random on a $10\,\text{m} \times 10\,\text{m}$ field. They perform a synchronized fusion step every 5th time step. Since the most recent five measurements are hence uncorrelated, a square-root decomposition of the measurement noise is not needed as only older measurements are correlated and their influence is increasingly becoming weaker. Therefore, we will utilize the additional information about uncorrelated measurements for the fusion. A summary of all used abbreviations and parameterizations of the used methods can be found in Table III. Fig. 5(a) shows the average mean squared error over all 25 sensor nodes. The time horizon for keeping track of process noise is $\mathcal{T}_Q = 5$. The results of the partial bounding SqDF_{Rbp} and the square-root decomposition without accounting for correlated measurements SqDF_{Qb} have the lowest MSE. As expected, the partial bounding SqDF_{Rbp} is more conservative than SqDF_{Qb} as indicated by the ANEES (see Fig. 5(b)). We also observe that SqDF_{Qb} is even consistent, i.e., the ANEES is close to one, which can be due to correlations that cancel each other out because of symmetries within the considered setup.

We also compared our proposed algorithm to the channel filter (ChF) [21], [41], which can be seen as a first-order approximation of the information graph technique. While the channel filter is suboptimal because it does not account for all common information, it might be only slightly suboptimal if the time between the occurrence of correlated estimation errors and the current fusion step is large enough. Further, it requires very little additional computation and communication.

Fig. 5 shows that the channel filters mean squared error is very close to the fusion result using CI. Yet, the ANEES indicates that the fusion result is consistent.

6. Results and Discussion

The second example shows that the convergence rate is improved when cross-covariances can be reconstructed accurately. However, the fusion can lead to numerical issues when sensor nodes are highly correlated since the joint covariance matrix cannot be inverted properly. While the additional square-root decomposition of the measurement noise is beneficial, it leads to additional communication that grows with the number of sensor nodes. It might be possible to discard parts of these square-roots when they traveled too far from their source. Therefore, correlations would only be tracked within a particular region of interest around a sensor node, which might improve the scalability of the algorithm. The choice of the time horizon determining the number of encoded dependent noise terms highly depends on the application and needs thorough consideration.

7. Conclusion

This paper aims at solving the problem of fusing multiple state estimates in different network topologies with unknown correlations. The proposed method utilizes the square-root decomposition of correlated noise covariances. The advantage of this approach is that every node can keep track of its local processing steps independently, and thus, no dedicated fusion center is necessary to manage the sensor nodes or their communication with each other.

The results show that the fused estimate remains consistent in arbitrary network topologies and that the fusion results of several sensor nodes converge faster towards a consensus than other fusion methods while yielding more consistent results than consensus methods. The downside of this method is the increased amount of additional information that needs to be shared and constantly updated. Therefore, the choice of the right time horizon might be crucial for the performance in many applications.

The findings of this paper make several contributions to the current literature. First, the modification of cross-covariances between state estimates due to intermediate fusion steps is discussed. Second, the additional dependency due to the double-counting of measurement information is examined. The provided method can be tailored to the needs of the application by tuning the time horizon for the number of tracked correlated noise covariances to meet the bandwidth requirements. Further, the time horizon for common prior information and common process noise can be chosen independently from the time horizon for common measurement information. This allows to only keep track of correlated estimation errors that contribute the most to the cross-covariance. By choosing a shorter time horizon, the fusion result becomes more conservative while still being a tighter bound than most other conservative fusion methods. In addition, the time horizon can be adjusted within the sensor network to provide more accuracy in certain areas where it is needed while allowing for rougher estimates in others.

Because of its flexibility, the method can even be utilized in low-cost sensor networks. An exciting application is the cooperative localization of robots, where many sources of correlated estimation errors occur, which are usually only addressed in a conservative fashion.

References

[1] S. Radtke, B. Noack, and U. D. Hanebeck, "Fully Decentralized Estimation Using Square-Root Decompositions," in *Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020)*, Virtual, Jul. 2020, pp. 1–8.

[2] D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, Eds., *Distributed Data Fusion for Network-Centric Operations*, ser. The Electrical Engineering and Applied Signal Processing Series. Boca Raton, Florida, USA: CRC Press, 2013.

[3] S. Grime and H. F. Durrant-Whyte, "Data Fusion in Decentralized Sensor Networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, Oct. 1994.

[4] C. Chong, K. Chang, and S. Mori, "A Review of Forty Years of Distributed Estimation," in *Proceedings of the 21st International Conference on Information Fusion (Fusion 2018)*, Cambridge, United Kingdom, Jul. 2018, pp. 1–8.

[5] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton, Florida, USA: CRC Press, 1998.

[6] A. G. O. Mutambara and H. F. Durrant-Whyte, "Nonlinear Information Space: A Practical Basis for Decentralization," in *Proceedings of SPIE's International Symposium on Photonics for Industrial Applications, Sensor Fusion VII*, P. S. Schenker, Ed., vol. 2355, no. 1. SPIE, 1994, pp. 97–105.

[7] B. Noack, D. Lyons, M. Nagel, and U. D. Hanebeck, "Nonlinear Information Filtering for Distributed Multisensor Data Fusion," in *Proceedings of the 2011 American Control Conference (ACC 2011)*, San Francisco, California, USA, Jun. 2011, pp. 4846–4852.

[8] J. L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, Apr. 1979.

[9] F. Govaers and W. Koch, "An Exact Solution to Track-to-track Fusion at Arbitrary Communication Rates," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2718–2729, Jul. 2012.

[10] M. Reinhardt, B. Noack, and U. D. Hanebeck, "The Hypothesizing Distributed Kalman Filter," in *Proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2012)*, Hamburg, Germany, Sep. 2012, pp. 305–312.

[11] Y. Bar-Shalom and L. Campo, "On the Track-to-Track Correlation Problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, Apr. 1981.

[12] C.-Y. Chong, K.-C. Chang, and S. Mori, "Distributed Tracking in Distributed Sensor Networks," in *Proceedings of the 1986 American Control Conference (ACC 1986)*, Seattle, Washington, USA, 1986, pp. 1863–1868.

[13] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion," in *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, H. Fourati, Ed. Boca Raton, Florida, USA: CRC Press, Aug. 2015, pp. 169–192.

[14] Y. Bar-Shalom and L. Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance," in *IEEE Transactions on aerospace and electronic systems*, vol. Vol. AES-22. No. 6, 1986, pp. 803–805.

[15] C.-Y. Chong and S. Mori, "Convex Combination and Covariance Intersection Algorithms in Distributed Fusion," in *Proceedings of the 4th International Conference on Information Fusion (Fusion 2001)*, Montréal, Québec, Canada, Aug. 2001.

[16] S. J. Julier and J. K. Uhlmann, "A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations," in *Proceedings of the IEEE American Control Conference (ACC 1997)*, vol. 4, Albuquerque, New Mexico, USA, Jun. 1997, pp. 2369–2373.

[17] L. Chen, P. O. Arambel, and R. K. Mehra, "Fusion under Unknown Correlation: Covariance Intersection Revisited," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1879–1882, Nov. 2002.

[18] M. Reinhardt, B. Noack, P. O. Arambel, and U. D. Hanebeck, "Minimum Covariance Bounds for the Fusion under Unknown Correlations," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1210–1214, Sep. 2015.

[19] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Decentralized Data Fusion with Inverse Covariance Intersection," *Automatica*, vol. 79, pp. 35–41, May 2017.

[20] B. Noack, J. Sijs, and U. D. Hanebeck, "Inverse Covariance Intersection: New Insights and Properties," in *Proceedings of the 20th International Conference on Information Fusion (Fusion 2017)*, Xi'an, China, Jul. 2017, pp. 1–8.

[21] C.-Y. Chong, K.-C. Chang, and S. Mori, "Fundamentals of Distributed Estimation," in *Distributed Data Fusion for Network-Centric Operations*, ser. The Electrical Engineering and Applied Signal Processing Series, D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, Eds. Boca Raton, Florida, USA: CRC Press, 2013, pp. 95–124.

[22] C.-Y. Chong, "Hierarchical Estimation," in *MIT/ONR Workshop on C3 Systems*, Monterey, California, USA, 1979, pp. 205–220.

[23] X. Tian, Y. Bar-Shalom, T. Yuan, E. Blasch, K. Pham, and G. Chen, "A generalized information matrix fusion based heterogeneous track-to-track fusion algorithm," in *Signal Processing, Sensor Fusion, and Target Recognition XX*, I. Kadar, Ed., vol. 8050, International Society for Optics and Photonics. SPIE, 2011, pp. 469 – 476.

[24] R. Olfati-Saber, "Distributed Kalman Filter with Embedded Consensus Filters," in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2005)*, Sevilla, Spain, Dec. 2005, pp. 8179–8184.

[25] G. Battistelli and L. Chisci, "Kullback-Leibler Average, Consensus on Probability Densities, and Distributed State Estimation with Guaranteed Stability," *Automatica*, vol. 50, no. 3, p. 707718, Mar. 2014.

[26] R. Olfati-Saber, "Distributed Kalman Filtering for Sensor Networks," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, New Orleans, Louisiana, USA, Dec. 2007, pp. 5492–5498.

[27] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-Based Linear and Nonlinear Filtering," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1410–1415, May 2015.

[28] J. Liu, Y. Liu, K. Dong, Z. Ding, and Y. He, "A Novel Distributed State Estimation Algorithm with Consensus Strategy," *Sensors*, vol. 19, no. 9, pp. 1–29, 2019.

[29] C.-Y. Chong, K.-C. Chang, and S. Mori, "Comparison of Optimal Distributed Estimation and Consensus Filtering," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, 2016, pp. 1034–1041.

[30] J. Čurn, D. Marinescu, N. O'Hara, and V. Cahill, "Data Incest in Cooperative Localisation with the Common Past-Invariant Ensemble Kalman Filter," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013, pp. 68–76.

[31] M. Reinhardt, B. Noack, and U. D. Hanebeck, "Reconstruction of Joint Covariance Matrices in Networked Linear Systems," in *Proceedings of the 48th Annual Conference on Information Sciences and Systems (CISS 2014)*, Princeton, New Jersey, USA, Mar. 2014, pp. 1–6.

[32] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck, "Optimal Sample-Based Fusion for Distributed State Estimation," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, Jul. 2016, pp. 1600–1607.

[33] S. Radtke, B. Noack, U. D. Hanebeck, and O. Straka, "Reconstruction of Cross-Correlations with Constant Number of Deterministic Samples," in *Proceedings of the 21st International Conference on Information Fusion (Fusion 2018)*, Cambridge, United Kingdom, Jul. 2018, pp. 1638–1645.

[34] M. Reinhardt, S. Kulkarni, and U. D. Hanebeck, "Generalized Covariance Intersection based on Noise Decomposition," in *Proceedings of the 2014 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2014)*, Beijing, China, Sep. 2014, pp. 1–8.

[35] S. Radtke, B. Noack, and U. D. Hanebeck, "Distributed Estimation using Square Root Decompositions of Dependent Information," in *Proceedings of the 22nd International Conference on Information Fusion (Fusion 2019)*, Ottawa, Canada, Jul. 2019, pp. 1–8.

[36] X. R. Li, Yunmin Zhu, Jie Wang, and Chongzhao Han, "Optimal Linear Estimation Fusion – Part I: Unified Fusion Rules," *IEEE Transactions on Information Theory*, vol. 49, no. 9, pp. 2192–2208, 2003.

[37] K.-C. Chang, R. K. Saha, and Y. Bar-Shalom, "On Optimal Track-to-Track Fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, Oct. 1997.

[38] W. Niehsen, "Information Fusion based on Fast Covariance Intersection Filtering," in *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*, Annapolis, Maryland, USA, Jul. 2002, pp. 901–904 vol.2.

[39] X. R. Li, Z. Zhao, and V. P. Jilkov, "Practical Measures and Test for Credibility of an Estimator," in *Proceedings of Workshop on Estimation, Tracking, and Fusion - A Tribute to Yaakov Bar-Shalom*, 2001, pp. 481–495.

[40] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-Based Linear and Nonlinear Filtering," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1410–1415, 2015.

[41] K. Chang, C. Chong, and S. Mori, "Analytical and Computational Evaluation of Scalable Distributed Fusion Algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 4, pp. 2022–2034, 2010.

Susanne Radtke received her master's degree in Mechatronics from the Technical University of Ilmenau, Germany, in 2016. Since 2017 she is a Ph.D. student at the Intelligent SensorActuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. Her research interests include distributed estimation and data fusion in sensor networks.

Benjamin Noack received his diploma in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2009. Afterwards, he obtained his Ph.D. in 2013 at the Intelligent SensorActuator- Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. Since 2013 he is a senior researcher at the Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the areas of multi-sensor data fusion, distributed and decentralized Kalman filtering, combined stochastic and set-membership approaches to state estimation, and event-based systems.

Uwe D. Hanebeck is a chaired professor of Computer Science at the Karlsruhe Institute of Technology (KIT) in Germany and director of the Intelligent SensorActuator-Systems Laboratory (ISAS). He obtained his Ph.D. degree in 1997 and his habilitation degree in 2003, both in Electrical Engineering from the Technical University in Munich, Germany. His research interests are in the areas of information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models. He is author and coauthor of more than 500 publications in various high-ranking journals and conferences and an IEEE Fellow.