Reconstruction of Cross-Correlations with Constant Number of Deterministic Samples

Susanne Radtke*, **Benjamin Noack***, **Uwe D. Hanebeck***, and **Ondřej Straka**‡

* 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

‡European Centre of Excellence
New Technologies for Information Society and Department of Cybernetics
University of West Bohemia, Czech Republic
straka30@kky.zcu.cz

***Abstract*—Optimal fusion of estimates that are computed in a distributed fashion is a challenging task. In general, the sensor nodes cannot keep track of the cross-correlations required to fuse estimates optimally. In this paper, a novel technique is presented that provides the means to reconstruct the required correlation structure. For this purpose, each node computes a set of deterministic samples that provides all the information required to reassemble the cross-covariance matrix for each pair of estimates. As the number of samples is increasing over time, a method to reduce the size of the sample set is presented and studied. In doing so, communication expenses can be reduced significantly, but approximation errors are possibly introduced by neglecting past correlation terms. In order to keep approximation errors at a minimum, an appropriate set size can be determined and a trade-off between communication expenses and estimation quality can be found.**

I. Introduction

With the increasing amount of available sensor data, the need for decentralized processing [1], [2] is becoming more important. While centralized fusion also includes efficient implementations like the information filter [3]–[5] or the optimally distributed Kalman filter [6]–[8] and is able to yield optimal results, it lacks scalability, modularity, and robustness of decentralized sensor networks. Especially in applications with limited bandwidth, communication rate, and energy, distributing the workload over the network can be an alternative to the centralized approach. These advantages unfortunately come with the need for more sophisticated algorithms to deal with possible double counting of information and cross-correlations between state estimates [9]–[11].

A method to optimally fuse two estimates if the underlying cross-correlations are known or can be reconstructed is proposed in [12], which can be generalized to the simultaneous fusion of multiple estimates [13]. Unfortunately, keeping track of cross-correlations between sensor nodes is cumbersome and often infeasible in multi-sensor systems unless special network and communication topologies are used [14], [15]. Therefore, a variety of suboptimal fusion methods can be named that address the issue of unknown cross-correlations. Covariance intersection [16]–[18] makes no assumption about the correlations between sensor nodes and combines the local state estimate into a conservative fusion result. As covariance intersection is often too pessimistic, less conservative fusion strategies have been proposed like ellipsoidal intersection [19], [20], its further development inverse covariance intersection [21], [22], or improved parameterizations [23]–[25]. Also, approaches are studied to further tighten the bounds on the covariance matrices [26], [27].

The conservatism of suboptimal fusion methods states the reason why effort has been put into developing methods for reconstructing the actual cross-correlations. In [28], the problem of vehicle localization has been explored. The common-past-invariant-Ensemble-KF (CPI-EnKF), which is a generalization of the EnKF, has been used to calculate the cross-correlations between vehicles to be tracked. Based on this, [29] describes a representation of estimation errors with stochastic samples to reconstruct the joint covariance matrix. The samples are designed to take correlated and uncorrelated noise terms into account. However, these approaches are able to retrieve the actual cross-correlations only asymptotically with increasing number of samples.

Finally, [30] presents a novel method which utilizes deterministic samples to optimally fuse estimates. This method keeps and processes a small set of deterministic samples that is sufficient to represent cross-correlations exactly. Those samples are kept in parallel to the local state estimate and are growing in number with the passing time steps since initialization. State estimates, covariance matrices, and corresponding samples can then be sent to a fusion center where the actual cross-correlation can be reconstructed and the optimal result can be obtained.

In this paper, we propose a novel technique to reconstruct cross-correlations between sensor node estimates by keeping a limited set of deterministic samples. These samples therefore represent the history of the latest cross-correlations while

representing a concentrated form of information to enable resource-saving communication over a distributed network.

This paper is structured in the following way: Section II will focus on the fusion method and the analytic calculation of cross-correlations. Section III will recapitulate the Sample Fusion Method described in [30]. In Section IV a novel technique to overcome the limitations of the Sample Fusion method will be presented. Finally, Section V will evaluate the proposed method and compare it to other methods.

II. Formulation of Problem

The temporal development of the system is described by a discrete-time linear time-variant stochastic dynamic system

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

with state matrix $\mathbf{A}_k$, state vector $\underline{x}_k$, input matrix $\mathbf{B}_k$, input vector $\underline{u}_k$, and zero-mean white Gaussian system noise $\underline{w}_k$ with covariance matrix $\mathbf{Q}_k$. In a network, a number of L sensor nodes receive information about the observed system via a noisy linear measurement model

$$y_k^{(i)} = \mathbf{H}_k^{(i)} \underline{x}_k^{(i)} + \underline{v}_k^{(i)} \text{ with } \underline{v}_k \sim \mathcal{N}\big(\underline{0}, \mathbf{R}_k^{(i)}\big)\,,$$

where $\mathbf{H}_k^{(i)}$ describes the observation model and $\underline{v}_k^{(i)}$ the sensor noise with covariance $\mathbf{R}_k^{(i)}$. Every sensor node processes a local state estimate $\underline{x}_k^i$ and covariance matrix $\mathbf{P}_{k|k}^i$ using the Kalman Filter or one of its derivatives (e.g., EKF, UKF). To enhance the accuracy of the state estimate, the local state estimates and covariance matrices are fused by utilizing the Bar-Shalom/Campo formulas for the multi-sensor case [31].

With the joint mean vector

$$\underline{\hat{m}}_{k|k} = \left[\big(\underline{\hat{x}}_{k|k}^{(1)}\big)^\top \quad , \ldots , \quad \big(\underline{\hat{x}}_{k|k}^{(L)}\big)^\top \right]^\top \tag{1}$$

and joint covariance matrix

$$\mathbf{J}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k}^{(1)} & \mathbf{P}_{k|k}^{(1,2)} & \ldots & \mathbf{P}_{k|k}^{(1,L)} \\ \mathbf{P}_{k|k}^{(2,1)} & \mathbf{P}_{k|k}^{(2)} & \ldots & \mathbf{P}_{k|k}^{(2,L)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k|k}^{(L,1)} & \mathbf{P}_{k|k}^{(L,2)} & \ldots & \mathbf{P}_{k|k}^{(L)} \end{bmatrix}, \tag{2}$$

the fused estimates and covariance matrices can be calculated according to

$$\mathbf{P}_{k|k} = \Big(\mathbf{H}^\top \big(\mathbf{J}_{k|k}\big)^{-1} \mathbf{H} \Big)^{-1}, \tag{3}$$

$$\underline{\hat{x}}_{k|k} = \mathbf{P}_{k|k} \mathbf{H}^\top \big(\mathbf{J}_{k|k}\big)^{-1} \underline{\hat{m}}_{k|k}\,, \tag{4}$$

with matrix $\mathbf{H} = [\mathbf{I}, \ldots, \mathbf{I}]^\top$ containing the identity matrix $\mathbf{I}$ of the system state dimension. The block entries on the main diagonal $\mathbf{P}_{k|k}^{(1)} \ldots \mathbf{P}_{k|k}^{(L)}$ account for the local covariance matrices of the sensor nodes and can be easily obtained. By contrast, the remaining entries account for the cross-correlations between the sensor nodes and are usually unknown. Neglecting cross-correlations and therefore assuming sensor nodes to be uncorrelated will yield erroneous and too optimistic fusion results. Two major reasons for correlations between sensor nodes can be distinguished:

- double counting of common measurement information,
- common system noise and prior information.

In this paper, we will mainly be concerned with the issue of common system noise and prior information. It occurs due to the fact that all sensor nodes use the same linear process model including the same system noise. This issue becomes more visible when calculating the cross-covariance matrix of the predicted state estimates [11]

$$\begin{aligned} \mathbf{P}_{k|k-1}^{(i,j)} &= E[(\underline{\hat{x}}_{k|k-1}^{(i)} - \underline{x}_k)(\underline{\hat{x}}_{k|k-1}^{(j)} - \underline{x}_k)^T] \\ &= \mathbf{A}_k \mathbf{P}_{k-1|k-1}^{(i,j)} \mathbf{A}_k^T + \mathbf{Q}_k\,. \end{aligned} \tag{5}$$

As a next step, the cross-correlation is additionally altered by the filtering step by

$$\begin{aligned} \mathbf{P}_{k|k}^{(i,j)} &= E[(\underline{\hat{x}}_{k|k}^{(i)} - \underline{x}_k)(\underline{\hat{x}}_{k|k}^{(j)} - \underline{x}_k)^T] \\ &= (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \mathbf{P}_{k|k-1}^{(i,j)} (\mathbf{I} - \mathbf{K}_k^{(j)} \mathbf{H}_k^{(j)})^T\,. \end{aligned} \tag{6}$$

Formulas (5) and (6) show the recursive calculation of the cross-covariance matrix. It is obvious, that in order to keep track of the cross-correlation between sensor nodes, an enormous amount of data (e.g., $\mathbf{A}_k$,$\mathbf{H}_k$,$\mathbf{K}_k$,$\mathbf{Q}_k$ for several time steps k) has to be saved and distributed over the network. A more dense representation of this history would therefore reduce the amount of data and enable distributed estimation.

III. Optimal Sample-Based Fusion for Linear Systems

In this section, we explain the Sample Fusion method described in [30]. To reconstruct the cross-correlation between sensor nodes the Sample Fusion method uses deterministic samples. The samples are obtained using the simple deterministic spherical simplex sampling method described in [32].

For an N-dimensional system with W-dimensional system noise $\underline{w}_k$ the dimension D of the created simplex is $D = N + P \times W$, where P accounts for a user-defined number of processing steps that are incorporated in the samples. All nodes create the *identical* set $\{\underline{p}^{(m)}\}_{m=1}^M$ of $M = D + 1$ equally weighted samples with sample mean zero.

In contrast to [30], we propose a simplification of the creation of samples by neglecting the weighting factor w during initialization. Therefore we obtain the sample covariance matrix according to

$$\sum_{m=1}^{M} \underline{p}^{(m)} = 0\,, \ \sum_{m=1}^{M} \underline{p}^{(m)} (\underline{p}^{(m)})^\top = \mathbf{I}_{D \times D}$$

which is equal to the identity. This set of samples is now initialized at time step k and weighted by the matrix $\mathbf{D}_k$

$$\mathbf{D}_k = \operatorname{diag}(\mathbf{P}_{k|k}, \mathbf{Q}_{k+1}, \ldots, \mathbf{Q}_{k+P})\,, \tag{7}$$

where $\mathbf{P}_{k|k}$ accounts for the known cross-correlation between the sensor nodes during initialization and $\mathbf{Q}_{k+1} \ldots \mathbf{Q}_{k+P}$ accounts for the process noise from initialization time step k to the time horizon P. By Cholesky decomposition of $\mathbf{D}_k =$

$\mathbf{L}_k \mathbf{L}_k{}^\top$, which can be seen as a Cholesky decomposition of the block entries on the main diagonal

$$\mathbf{L}_k = \operatorname{diag}\big(\mathbf{p}_{k|k}, \mathbf{q}_{k+1}, \ldots, \mathbf{q}_{k+P}\big)$$

the sample set can be weighted according to

$$\begin{aligned}\underline{d}_k^{(m)} &= \mathbf{L}_k \underline{p}^m \ , \ \ \forall m = 1, \ldots, M \\ &= \Big[(\underline{s}_{k|k}^{i,m})^\top, \quad (\underline{w}_{k+1}^{(m)})^\top \quad , \ldots, \quad (\underline{w}_{k+P}^{(m)})^\top \Big]^\top . \quad (8)\end{aligned}$$

By using the special characteristics of the simple deterministic spherical simplex sampling method the elements of matrix $\mathbf{D}_k$ can be reconstructed by multiplying the elements of $\underline{d}_k^{(m)}$ referring to the same correlation matrix

$$\mathbf{P}_{k|k} = \sum_{m=1}^{M} \underline{s}_{k|k}^{(i,m)} \big(\underline{s}_{k|k}^{(i,m)}\big)^\top ,$$

$$\mathbf{Q}_{k+P} = \sum_{m=1}^{M} \underline{w}_{k+P}^{(i,m)} \big(\underline{w}_{k+P}^{(i,m)}\big)^\top$$

while multiplying heterogeneous elements of the set yields the zero matrix

$$\underline{\mathbf{0}} = \sum_{m=1}^{M} \underline{s}_{k|k}^{(i,m)} \big(\underline{w}_{k+P}^{(m)}\big)^\top$$

meaning that they are fully uncorrelated. This makes sense as the process noise is white Gaussian noise and uncorrelated to the initial correlation of the sensor nodes and also uncorrelated to process noise terms at different time steps.

During the time update of the k-th time step, the first sample set $\underline{s}_{k-1|k-1}^{(i,m)}$ will be propagated according to the system model and then the corresponding sample representing the process noise $\underline{w}_k^{(m)}$ will be added

$$\underline{s}_{k|k-1}^{(i,m)} = \mathbf{A}_k \underline{s}_{k-1|k-1}^{(i,m)} + \underline{w}_k^{(m)} \ , \ \ \forall m = 1, \ldots, M \, .$$

The measurement update is performed, yielding

$$\underline{s}_{k|k}^{(i,m)} = (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \mathbf{s}_{k|k-1}^{(i)} \ , \ \ \forall m = 1, \ldots, M \, .$$

In case no measurement is available, the prediction is kept, respectively $\underline{\hat{x}}_{k|k}^{(i)} = \underline{\hat{x}}_{k|k-1}^{(i)}$, $\underline{\mathbf{P}}_{k|k}^{(i)} = \underline{\mathbf{P}}_{k|k-1}^{(i)}$ and $\underline{s}_{k|k}^{(i)} = \underline{s}_{k|k-1}^{(i)}$. After P or fewer time steps, the cross-correlation can be reconstructed using

$$\mathbf{P}_{k+P|k+P}^{i,j} = \sum_{m=1}^{M} \underline{s}_{k+P|k+P}^{(i,m)} \big(\underline{s}_{k+P|k+P}^{(j,m)}\big)^\top . \quad (9)$$

After the time horizon P is reached, no further process noise matrices can be included in the samples to account for newer processing steps. In this case, either new noise samples can be added or the fusion has to be executed. Afterwards, the system is reinitialized with the fused covariance matrix $\mathbf{P}_{P|P}$ and the sample creation starts from the beginning.

IV. Restricting the Number of Deterministic Samples

As the time interval since the initialization of the system increases, the number of samples is growing linearly with the number of time steps P. A large number of samples also means higher costs for propagating the samples through the network and calculating the joint covariance matrix.

This paper proposes to keep a limited set of samples and neglect all data older than a predefined time horizon p_{max}. This is possible, because the samples are weighted every measurement step with $\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}$. Therefore, the samples are approaching zero after a sufficient time for $\mathbf{K}_k^{(i)} > \mathbf{0}$ depending on the system settings. Additionally, the system matrix $\mathbf{A}_k$ can support this development if the system is asymptotically stable.

Because of the nested structure of the samples, a later extraction of data from the samples is impossible or has to be done in all nodes simultaneously. Since this is not feasible in distributed systems, we propose to keep a set of characteristic parameters $\underline{\mathbf{T}}$ to keep a history of the latest processing steps. These parameters can then be used to initialize a new set of samples which can be distributed over the network.

A. Sample-creation with a set of characteristic parameters

In the case of linear or linearized systems, we can easily obtain the matrix transformation to calculate the cross-correlations between sensor nodes. This matrix transformation can be seen in (5) and (6) as discussed in Section II

$$\begin{aligned}\mathbf{P}_{k|k}^{(i,j)} &= (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \mathbf{A}_k \mathbf{P}_{k-1|k-1}^{(i,j)} \mathbf{A}^\top (\mathbf{I} - \mathbf{K}_k^{(j)} \mathbf{H}_k^{(j)})^\top \\ &\quad + (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \mathbf{Q}_k (\mathbf{I} - \mathbf{K}_k^{(j)} \mathbf{H}_k^{(j)})^\top .\end{aligned}$$

This can be rewritten using characteristic parameters $\mathbf{T}_0$ and $\mathbf{T}_1$ to account for the executed matrix transformations

$$\mathbf{P}_{k|k}^{i,j} = \mathbf{T}_1^i \mathbf{P}_{k-1|k-1}^{i,j} \mathbf{T}_1^j + \mathbf{T}_0^i \mathbf{Q}_k \mathbf{T}_0^j \, .$$

This set of characteristic parameters will be saved locally at the sensor nodes and updated every time step. The parameters will grow linearly to a number of $M = p_{max} + 1$ matrices, where p_{max} is a predefined user setting on how many time steps are included in the samples. Once the maximum number is reached the oldest parameters will be discarded. Hence, only a history of the latest matrix transformations will be stored

$$\begin{aligned}\mathbf{P}_{k+P|k+P}^{(i,j)} &= \mathbf{T}_P^{(i)} \mathbf{P}_{k-P|k-P}^{(i,j)} {\mathbf{T}_P^{(j)}}^\top + \mathbf{T}_{P-1}^{(i)} \mathbf{Q}_k {\mathbf{T}_{P-1}^{(j)}}^\top \\ &\quad + \cdots + \mathbf{T}_0^{(i)} \mathbf{Q}_{k+P} {\mathbf{T}_0^{(j)}}^\top .\end{aligned}$$

The characteristic parameters can be calculated recursively. To begin with, the set of parameters is initialized with $\mathbf{T}_0 = \mathbf{I}_{n \times n}$, while the rest of the set is initialized with the zero matrix. At every time step, the parameters of the last time step are shifted. By that, the oldest parameter $\mathbf{T}_p$ is discarded and overwritten. Simultaneously, a new parameter $\mathbf{T}_0$ is initialized

to account for the matrix transformation applied to the newest process noise matrix. Therefore, at every time step we obtain

$$\mathbf{T}_{m+1}^{(i)} = \mathbf{T}_{m}^{(i)} \ , \ \forall m = 0, \dots, P-1 \, , \tag{10}$$

$$\mathbf{T}_{m}^{(i)} = (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \mathbf{A}_k \mathbf{T}_{m+1}^{(i)}, \forall m = 0, \dots, P \, , \tag{11}$$

$$\mathbf{T}_{0}^{(i)} = (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \, . \tag{12}$$

The matrix $\mathbf{D}_k$ referred to in equation (7) can be rewritten as

$$\mathbf{D}_k = \mathrm{diag}(\mathbf{\Sigma}_0, \mathbf{\Sigma}_1 \dots, \mathbf{\Sigma}_{k+P}). \tag{13}$$

with the generalized matrices $\mathbf{\Sigma}_0 \dots \mathbf{\Sigma}_{k+P}$. The matrix is initialized with $\mathbf{\Sigma}_0 = \mathbf{P}_{k-1|k-1}$ while the rest of the matrices are filled with the zero matrix. The matrices are then shifted in the same way as the characteristic parameters. Hence, the last matrix $\mathbf{\Sigma}_{k+P}$ which contains the latest cross-correlation or process noise matrix is discarded and the newest process noise matrix is introduced by $\mathbf{\Sigma}_0$

$$\mathbf{\Sigma}_{m+1}^{(i)} = \mathbf{\Sigma}_{m}^{(i)} \ , \ \forall m = 0, \dots, P-1 \, , \tag{14}$$

$$\mathbf{\Sigma}_{0}^{(i)} = \mathbf{Q}_k \, . \tag{15}$$

At the fusion time step the matrix $\mathbf{D}_k$ will be Cholesky decomposed in $\mathbf{D}_k = \mathbf{L}_k {\mathbf{L}_k}^\top$ and then used to weight the identity sample set as described in equation (8). This results in a generalized sample set

$$\underline{d}_k^{(m)} = \mathbf{L}_k \underline{p}^{(m)} \ , \ \ \forall m = 1, \dots, M \tag{16}$$

$$= \left[(\underline{s}_0^{(m)})^\top, \quad (\underline{s}_1^{(m)})^\top \quad , \dots , \quad (\underline{s}_P^{(m)})^\top \right]^\top . \tag{17}$$

This sample set is then weighted with the characteristic parameters and summed up to obtain the sample set $s_{k+P|k+P}^{i,m}$ which includes the full history of all matrix transformations over the last P time steps

$$(\underline{s}_{k+P|k+P}^{i,m})^\top = \sum_{i=0}^{P} \mathbf{T}_i \underline{s}_i^{(m)}. \tag{18}$$

Afterwards the cross-covariance matrix can be reconstructed using (9). After the fusion is executed the system can additionally be reinitialized. With that, the local state estimate and covariance will be replaced by the fused results. Special care has to be taken to avoid double counting in this case because the new estimate will be fully correlated to the state estimates incorporated. This full correlation is not reflected in the current characteristic parameter set $\underline{\mathbf{T}}$ and thus has to be calculated from the received sample set. Since this is not a trivial calculation reinitializing the characteristic parameter set $\underline{\mathbf{T}}$ and taking care of the time of the next fusion step can reduce the amount of data-incest. The next fusion step should be executed, after p_{max} time steps are processed assuming that this is a sufficient time horizon to deposit old processing steps.

A summary of the proposed algorithm can be found in Algorithm 1. In Section IV-D the belonging network topology is discussed.

Algorithm 1 Sample Fusion method with a reduced set of deterministic Samples

// Set initial state estimate $\hat{\underline{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ at every node
$\mathbf{\Sigma}_0 = \mathbf{P}_{0|0}^{(i,j)}, \ \mathbf{\Sigma}_{1\dots P} = \underline{\mathbf{0}}$
$\mathbf{T}_0 = \mathbf{I}, \quad \mathbf{T}_{1\dots P} = \underline{\mathbf{0}}$
for $k = 0, 1, 2, \dots$ **do**
    i-th node calculates time- and measurement-update
    // Update of $\underline{\mathbf{T}}$ and $\underline{\mathbf{\Sigma}}$
    for $m = 0, \dots, P-1$ **do**
        Shifting of $\underline{\mathbf{T}}$-elements (10)
        Shifting of $\underline{\mathbf{\Sigma}}$-elements (14)
        Update of $\underline{\mathbf{T}}$-elements (11)
    end for
    Creation of $\mathbf{T}_0$ (12)
    Creation of $\mathbf{\Sigma}_0$ (15)
    // Sample-Creation
    Create weighting Matrix $\mathbf{D}_k$ (13)
    Perform Cholesky-decomposition of $\mathbf{D}_k$ (16)
    Create identity sample set $\underline{p}^{(m)}$ (see [30])
    Create process noise sample set (17)
    Create cross correlation sample set $(\underline{s}_{k|k}^{(i,m)})^\top$ (18)
    // Fusion step
    if perform fusion **then**
        send: $\hat{\underline{x}}_{k|k}^{(i)}$,$\mathbf{P}_{k|k}^{(i)}$ and $(\underline{s}_{k|k}^{(i,m)})^\top$
        receive: $\underline{x}_k^{(j)}$,$\mathbf{P}_k^{(j)}$ and $(\underline{s}_{k|k}^{(j,m)})^\top$
        Calculate cross-correlation $\mathbf{P}_{k+P|k+P}^{(i,j)}$ (9)
        Fuse state and covariance $\hat{\underline{x}}_{k|k}$, $\mathbf{P}_{k|k}$ (1),(2),(3),(4)
    end if
    // Reinitializing with fusion results
    if reinit **then**
        $\hat{\underline{x}}_{k|k}^{(i)} = \hat{\underline{x}}_{k|k}$, $\mathbf{P}_{k|k}^{(i)} = \mathbf{P}_{k|k}$
        $\mathbf{\Sigma}_0 = \mathbf{P}_{k|k}, \ \mathbf{\Sigma}_{1\dots P} = \underline{\mathbf{0}}$
        $\mathbf{T}_0 = \mathbf{I}, \ \ \mathbf{T}_{1\dots P} = \underline{\mathbf{0}}$
    end if
end for

B. Extension to nonlinear systems

The advantage of using samples to reconstruct the cross-correlations is that they are a straightforward approach for nonlinear systems. Hence, using a set of characteristic parameters to account for linear transformations does not support this advantages at the first glance. Nonetheless, it can be stated that systems with slight nonlinearities that transform a Gaussian distribution into another Gaussian distribution can be described by linear transformations. These parameters can also be extracted for example from the UKF [33].

In systems with high nonlinearities, more thought has to be put into the choice of samples and their propagation.

C. Extension to systems with lower noise dimension

Another possible description of the temporal development of the system is

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

where $\mathbf{Q}_k$ can have a lower dimension then $\mathbf{A}_k$. Two possible solutions can be proposed. The first solution is to create noise samples with lower dimension while keeping the initial cross-correlation until the time horizon P is reached. Afterwards, the cross-correlation is discarded and only noise matrices with lower dimension are left to be processed. The characteristic parameter set $\underline{\mathbf{T}}$ is updated with

$$\mathbf{T}_m = (\mathbf{I} - \mathbf{K}_k^{(i)}\mathbf{H}_k^{(i)})\mathbf{B}_k \ , \ \forall m = 1, \dots P \ .$$

Because of the different sizes of the parameters and the entries in $\mathbf{D}_k$ this might lead to a higher implementation complexity while on the other hand consuming less memory and also creating fewer samples which will result in a lower bandwidth requirement. The second approach would be, to initialize the elements of matrix $\mathbf{D}_k$ with $\mathbf{\Sigma}_k = \mathbf{B}\mathbf{Q}_k\mathbf{B}^\top$. In this case, there is no need to account for different dimensions of parameters $\underline{\mathbf{T}}$ while on the downside the advantages of lower memory and sample size are not fully exploited.

D. Discussion

The proposed method could be used in a parallel bus-structure similar to Figure 1 which can be seen as a Broadcasting Topology [34]. Triggered for example by an external signal every node will send its state estimate, the covariance matrix and the sample set over the network at the same time step. The size of the sample set could be designed accordingly to be able to send this information within a single time step. Afterwards, every node will perform a fusion step and update its state estimate and covariance with the fusion result. The next fusion step should be performed after the time horizon for the sample set is passed. This will bypass the problem of full correlation between the nodes at the fusion time step because the information will then be already discarded and will not be reflected in the samples.

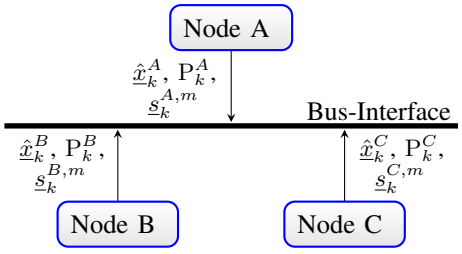
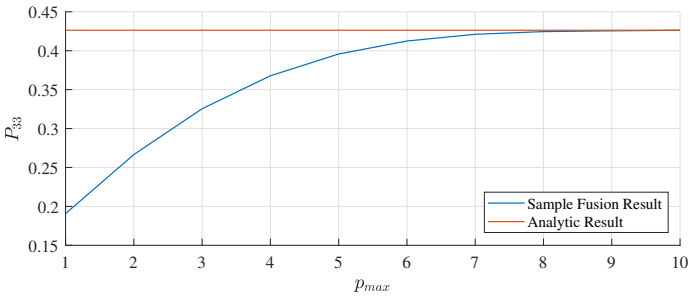


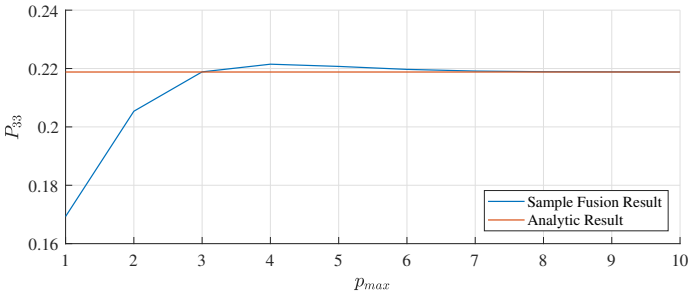
Fig. 1: Broadcasting Topology

V. Evaluation

The following section will present two evaluation examples to show the accuracy of the derived method. The first example will be a White-Noise Acceleration Model to show the dependency between the horizon p_{max} and the compliance with the optimal result. Furthermore, the resulting communication costs will be discussed. The second example will show a simple tracking problem where different fusion methods will be compared.



(a) with $\mathbf{R} = 0.1$



(b) with $\mathbf{R} = 0.005$

Fig. 2: Cross-correlation of state x_3 contained in $P_{k|k}^{A,B}$, comparison of result obtained analytically and from Sample Fusion with different time horizon p_{max}.

A. Example 1

In this section we will use a White-Noise Acceleration Model

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

$$\mathbf{A} = \begin{bmatrix} 1 & T & T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \ , \ \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ , \ \mathbf{Q} = 0.1$$

to show the dependency between the time horizon p_{max} and the compliance of the analytically obtained and the reconstructed cross-correlation. The two sensor nodes A and B employ a measurement model

$$\underline{y}_k^{(i)} = \mathbf{H}_k^{(i)}\underline{x}_k^{(i)} + \underline{v}_k \text{ with } \underline{v}_k \sim \mathcal{N}\big(\underline{0}, \mathbf{R}\big)$$

with measurements corrupted by additive-white Gaussian noise $\underline{v}_k$ with covariance matrix $\mathbf{R}$. The measurement noise is modeled as $\mathbf{R} = 0.1$ in Figure 2a and $\mathbf{R} = 0.005$ in Figure 2b. Further, sensor nodes A and B use different measurement matrices

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

The fusion is executed after 10 time steps. Figure 2 shows the cross-correlation of state x_3 between node A and B contained in the cross-correlation matrix $P_{k|k}^{A,B}$ at the fusion time step $k = 10$. It shows, that the variance reaches the analytically obtained value when the time horizon p_{max} is as big as the steps till fusion.

Further, the difference between Figure 2a and 2b shows that the convergence of the reconstructed to the analytically

TABLE I: Bandwidth Requirement

Algorithm	Data	Bandwidth Requirement ($N = 3$, $W = 1$, $P = 5$)
Naïve fusion, CI	$\underline{\hat{x}}$, $\mathbf{P}$	$3 + 9 = 12 = 96$Byte
Optimal Fusion	$\underline{\hat{x}}$, $\mathbf{P}$, $\mathbf{A}_{1\dots P}$ $\mathbf{H}_{1\dots P}$, $\mathbf{K}_{1\dots P}$ $\mathbf{Q}_{1\dots P}$, $\mathbf{B}_{1\dots P}$	$3 + 9$ $+5 \times (9 + 3 + 3 + 1 + 3)$ $= 107 = 856$Byte
Fusion with Parameter set $\underline{\mathbf{T}}$	$\underline{\hat{x}}$, $\mathbf{P}$, $\mathbf{B}_{1\dots P}$ $\mathbf{T}_0 \dots \mathbf{T}_P$,$\mathbf{Q}_{1\dots P}$	$3 + 9 + 9 + 5 \times 1 \times 3$ $= 86 = 688$Byte
Sample Fusion	$\underline{\hat{x}}$, $\mathbf{P}$ $\underline{s}^{(N+P\cdot W+1)}$	$3 + 9 + 3 \times (3 + 5 + 1)$ $= 39 = 312$Byte

obtained cross-correlations depends on the system properties. In this example, the cross-correlation approaches the analytic result faster with a lower measurement noise. The combination of system parameters leads to a lower impact of the process noise and therefore causes smaller errors when using fewer samples.

B. Communication cost

A fundamental problem of decentralized processing is the resource-efficient use of the limited bandwidth or communication rate. Reducing communication costs, therefore, enables the use of decentralized methods [35]. To demonstrate the communication costs of different methods we will use the assumptions of the 3-dimensional system utilized in Example 1 and also assume that the system is time variant and $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{Q}_k$ might change at every time step. For every matrix-entry or value a storage size of 8 Byte is used. For the memory consumption of the used parameters the following values are assumed: state estimate $\underline{\hat{x}} \triangleq N$, covariance matrix $\mathbf{P} \triangleq N^2$, state matrix $\mathbf{A}_k \triangleq N^2$, Kalman gain $\mathbf{K}_k \triangleq N$, observation model $\mathbf{H}_k \triangleq N$, process noise $\mathbf{Q}_k \triangleq W$, input matrix $\mathbf{B}_k \triangleq N$, with system dimension $N = 3$, process noise dimension $W = 1$, and time horizon $P = p_{max} = 5$.

Table I shows the found communication costs for four chosen fusion methods: naïve fusion, covariance intersection (CI), optimal Bar-Shalom/Campo fusion, fusion with characteristic parameters and Sample Fusion.

For the naïve approach, only the state and the covariance are transmitted. Clearly, the naïve approach consumes the least amount of data packages but will also yield the worst results. Covariance intersection (CI) can also be executed with only the state and the covariance matrix, but the method yields a better and more conservative result. For the optimal Bar-Shalom/Campo fusion, the complete information of all system matrices is needed to reconstruct the covariance matrix. Therefore this method requires the highest bandwidth. With the in Section IV derived characteristic parameters $\underline{\mathbf{T}}$ the data volume is lower due to the merging of the processing steps into the matrices. But still, the process noise matrix is needed to correctly reconstruct the cross-correlation. The Sample-Fusion method is the densest representation of the information and therefore requires the least bandwidth while still being able to correctly reconstruct the cross-correlation.

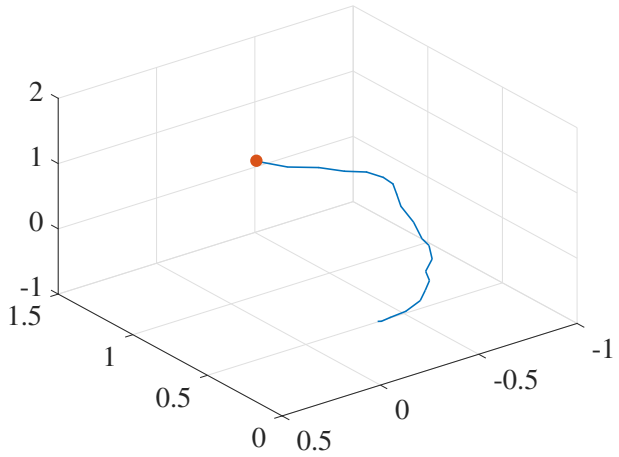


Fig. 3: Part of the non-linear trajectory (blue) of the tracked target (red).

C. Example 2

In this section, a simple track-to-track fusion is presented. Four cameras are observing independently from each other a moving object following a non-linear trajectory (see Figure 4). It is assumed that measurements are not available at every time step and therefore the local estimator is performing only a prediction in this case. This will lead to an increased covariance matrix and it is also hard to track for a fusion center. The number of time steps between each fusion event is set to $k_{\text{fusion}} = 10$ time steps.

The system state $\underline{x}_k$ is composed of the target position as well as its velocity. The temporal evolution is modeled by (compare with [30])

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

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & T\cdot\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} T\cdot\mathbf{I}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix}, \ \mathbf{Q} = 0.05\cdot\mathbf{I}_{3\times3}$$

where the time period T is $50\,\text{ms}$. The measurements will be corrupted by additive-white Gaussian noise $\underline{v}_k$ leading to the measurement model

$$\underline{y}_k^{(i)} = \mathbf{H}_k^{(i)}\underline{x}_k^{(i)} + \underline{v}_k.$$

In the first scenario, one of the nodes is chosen as fusion center and provides the user with a fused estimate. Furthermore the local estimators will continue working independently without being reinitialized with the fused estimate and covariance matrix.

In a second scenario, it is assumed that either a fusion center is providing all other sensor nodes with the fused estimate or that the fusion is completely distributed to all nodes, providing every node with the currently fused state estimate.

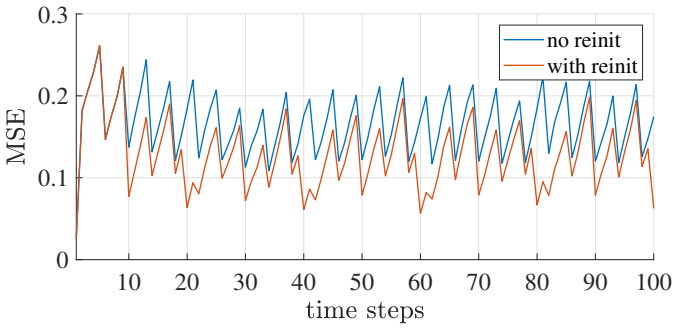


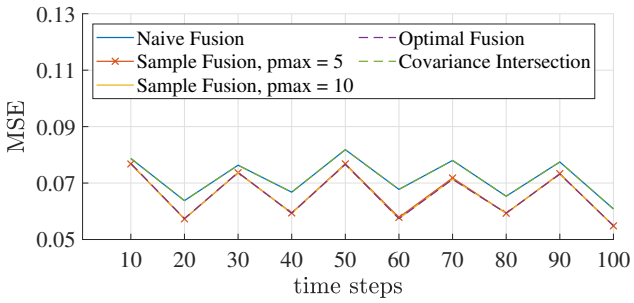
Fig. 4: Mean squared error (MSE) over 200 Monte Carlo runs, sensor nodes with measurements every 4th time step, without and with reinitialization with results obtained by Sample Fusion with $pmax = 10$.

We perform 200 Monte Carlo runs in which the object follows a non-linear trajectory. The results of both scenarios are compared with the estimates obtained by naïve fusion neglecting all cross-correlations between nodes, optimal fusion with full knowledge about the system and covariance intersection [16]–[18] which makes no assumptions about the cross-correlation and yields a conservative result. To compare these methods the mean squared error (MSE) between the state estimate and the real position of the target is calculated.

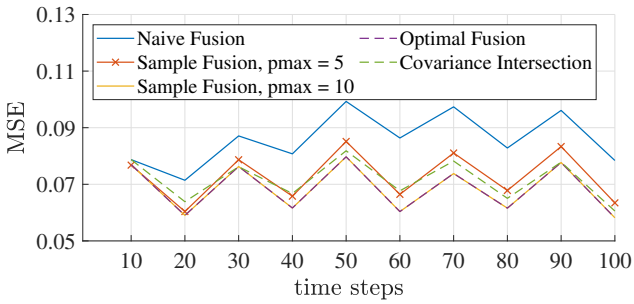
Figure 4 shows the MSE with and without reinitialization after the fusion result is obtained by the Sample Fusion method. The sensor nodes only receive a measurement every 4th time step resulting in a sawtooth curve. It is clearly visible, that the reinitialization of the sensor node with the fusion estimates and covariance results in a much lower mean squared error.

Figure 5 compares the naïve fusion, the optimal fusion, the covariance intersection and the Sample Fusion method with time horizon p_{max} equal to the fusion interval k_{fusion} and a lower time horizon. Figure 5a demonstrates the fusion without reinitialization. Only the naïve fusion and covariance intersection result in a higher MSE while the other methods perform equally good.

Figure 5b shows that an erroneous reinitialization will cause a decreased precision in estimation. Covariance intersection results in a conservative estimate and therefore causes smaller errors than naïve fusion. It can be shown that through reinitializing the system, the Sample Fusion method yields the exact same results as the Optimal Fusion when using the same time horizon as the fusion interval (see Figure 5b). This will also be the case if $\mathbf{Q}$ changes. Fewer samples result in an erroneous covariance matrix because the time horizon is not sufficient to include enough processing steps to reconstruct the full cross-correlation. The resulting reinitialization then results in larger errors than without reinitialization. The compliance between the optimal and the reconstructed cross-correlation depends on the ratio between the system noise $\mathbf{Q}_k$, the measurement noises $\mathbf{R}_k^{(1\dots L)}$ and the number of samples p_{max}.



(a) Scenario 1: Without reinitialization of state estimate and covariance matrix.



(b) Scenario 2: With reinitialization of state estimate and covariance matrix.

Fig. 5: Mean squared errors (MSE) over 200 Monte Carlo runs, comparison between four fusion methods: naïve Fusion, Optimal Fusion and Sample Fusion with $p_{max} = 10$ and $p_{max} = 5$.

VI. Conclusion

In this paper, we proposed a fusion method using a limited number of deterministic samples optimized for use in distributed state estimation. The obtained samples can either be sent to a fusion center or distributed to each node for decentralized fusion. The joint covariance matrix is then reconstructed by computing the sample cross-covariance matrices. Depending on the chosen time horizon, the samples only include the latest cross-correlations. The Sample Fusion method provides a very dense representation of the history of cross-correlations which is advantageous in networks with limited bandwidth or communication rate. Two evaluation examples show that depending on the horizon the samples can yield the optimal result for the fusion problem. Future work will have to investigate, how many samples have to be utilized to archive a sufficient representation of the cross-correlation. Further, a tight upper bound could be introduced to account for the cross-correlations not included in the current sample set.

Acknowledgment

This work was supported by the German Research Foundation (DFG) under grant HA 3789/14-1.

REFERENCES

[1] 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. CRC Press, 2013.

[2] C.-Y. Chong, "Forty Years of Distributed Estimation: A Review of Noteworthy Developments," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2017)*, Bonn, Germany, Oct. 2017.

[3] A. G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.

[4] 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.

[5] 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.

[6] 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.

[7] 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, Jul. 2012.

[8] 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.

[9] 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.

[10] 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.

[11] 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. CRC Press, Aug. 2015, pp. 169–192. [Online]. Available:

[12] 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.

[13] S.-L. Sun and Z.-L. Deng, "Multi-sensor Optimal Information Fusion Kalman Filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, Jun. 2004.

[14] C.-Y. Chong, "Hierarchical estimation," in *MIT/ONR Workshop on C3 Systems*, Monterey, California, USA, 1979.

[15] 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.

[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. [Online]. Available:

[19] J. Sijs and M. Lazar, "State-fusion with Unknown Correlation: Ellipsoidal Intersection," *Automatica*, vol. 48, no. 8, pp. 1874–1878, Aug. 2012.

[20] B. Noack, J. Sijs, and U. D. Hanebeck, "Algebraic Analysis of Data Fusion with Ellipsoidal Intersection," in *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sep. 2016.

[21] 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. [Online]. Available:

[22] 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.

[23] M. Reinhardt, B. Noack, and U. D. Hanebeck, "Closed-form Optimization of Covariance Intersection for Low-dimensional Matrices," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[24] U. Orguner, "Approximate Analytical Solutions for the Weight Optimization Problems of CI and ICI," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2017)*, Bonn, Germany, Oct. 2017.

[25] J. Ajgl, M. Šimandl, M. Reinhardt, B. Noack, and U. D. Hanebeck, "Covariance Intersection in State Estimation of Dynamical Systems," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[26] J. Ajgl and M. Šimandl, "On linear estimation fusion under unknown correlations of estimator errors," in *Proceedings of the 19th IFAC World Congress (IFAC 2014)*, Cape Town, South Africa, Aug. 2014.

[27] M. Zarei-Jalalabadi, S. M. B. Malaek, and S. S. Kia, "A Track-to-Track Fusion Method for Tracks With Unknown Correlations," *IEEE Control Systems Letter*, vol. 2, no. 2, pp. 189–194, Apr. 2018.

[28] 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.

[29] M. Reinhardt, B. Noack, and U. Hanebeck, "Reconstruction of Joint Covariance Matrices in Networked Linear Systems," in *2014 48th Annual Conference on Information Sciences and Systems, CISS 2014*, Mar. 2014.

[30] 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.

[31] V. Shin, Y. Lee, and T.-S. Choi, "Generalized Millman's formula and its application for estimation problems," *Signal Processing*, vol. 86, no. 2, pp. 257–266, Feb. 2006.

[32] S. J. Julier, "The Spherical Simplex Unscented Transformation," *Proceeding of the American Control Conference Denver Colorado June 4-6*, pp. 2430–2434, 2003.

[33] T. Lefebvre, H. Bruyninckx, and J. de Schutter, *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics. Springer, 2005, vol. 19.

[34] C.-Y. Chong, K.-C. Chang, and S. Mori, "Fundamentals of Distributed Estimation," in *Distributed Data Fusion for Network-Centric Operations*. CRC Press, 2013, pp. 95–124.

[35] N. G. Wah and Y. Rong, "Comparison of Decentralized Tracking Algorithms," *Proceedings of the Sixth International Conference of Information Fusion*, 2003.