Optimal Sample-Based Fusion for Distributed State Estimation

Jannik Steinbring*, **Benjamin Noack***, **Marc Reinhardt**†, and **Uwe D. Hanebeck***

*Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
jannik.steinbring@kit.edu, benjamin.noack@ieee.org, uwe.hanebeck@ieee.org

†IMC Financial Markets
Amsterdam, The Netherlands
marc.reinhardt@ieee.org

Abstract—**In this paper, we present a novel approach to optimally fuse estimates in distributed state estimation for linear and nonlinear systems. An optimal fusion requires the knowledge of the correct correlations between locally obtained estimates. The naive and intractable way of calculating the correct correlations would be to exchange information about every processed measurement between all nodes. Instead, we propose to obtain the correct correlations by keeping and processing a small set of deterministic samples on each node in parallel to the actual local state estimation. Sending these samples in addition to the local state estimate to the fusion center allows for correctly reconstructing the desired correlations between all estimates. In doing so, each node does not need any information about measurements processed on other nodes. We show the optimality of the proposed method by means of tracking an extended object in a multi-camera network.**

I. Introduction

Among other advantages, such as scalability, robustness, and increased spatial coverage, multisensor systems can significantly improve the estimation quality and performance over a single sensor. The optimal strategy to treat the acquired sensor data is a centralized processing of each measurement. Although efficient techniques, such as the information form [1] of the Kalman filter, can be named to pre-process and transmit a plethora of measurements to a data sink, such centralized estimation schemes are often not feasible. In particular, bandwidth limitations, connectivity changes, or energy consumption can render frequent communication intractable. In this regard, it is a major advantage of distributed estimation techniques that the processing load is distributed among the sensor nodes and data transmissions can take place at arbitrary rates. Each sensor node processes the locally acquired measurements and computes an estimate of the system's state. Consequently, the sensor nodes deliver state estimates in place of sensor readings to the data sink. In order to provide a global estimation result, the local estimates received by the data sink are fused into a single estimate, which is referred to as track-to-track fusion [2] in the context of distributed target-tracking applications.

Compared to a centralized processing of all measurements, distributed estimation techniques require more elaborate algorithms that take care of possible correlations between the local estimation errors. With the distributed Kalman filter [3], [4], it has been demonstrated that a local processing of measurements can be established such that the results can be combined after arbitrarily many time steps into an optimal estimate. More precisely, the optimal estimate is equivalent to the results of a centralized Kalman filter. However, the distributed Kalman filter is highly susceptible to any changes at both network and node level. In particular, each node must be aware of the other nodes' sensor models, scheduling policies, and noise parameters. Providing each node with full knowledge about the network is infeasible in many applications. In [5], [6], the required knowledge has been replaced by a hypothesis, which leads to suboptimal results at the data sink if the hypothesis deviates from the actual setup.

While the distributed Kalman filter constitutes a distributed implementation of a single Kalman filter, most distributed estimation systems consist of sensor nodes that are equipped with local Kalman filters. Each filter independently computes an optimal estimate given the local measurements. In general, the local processing and subsequent fusion of Kalman filter estimates does not yield the result of a single centralized Kalman filter. This means that an optimal fusion of local estimates is not equivalent to the centrally optimal estimate given all measurements. This discrepancy has been studied, for instance in [7], and can be addressed by a tracklet fusion method [8], [9], which extracts the new information from each local estimate compared to the last fusion result. However, either a full-rate communication to the data sink or an augmented state representation [10] is required in order to compute the optimal estimate and to prevent over-optimistic fusion results. In the latter case, joint state densities are employed to allow for irregular data transmissions.

The general problem when local estimates are to be fused resides in the lack of independence between their estimation errors. An optimal fusion result is only attainable if the correlation structure is known to the fusion system and can be exploited. The according algorithm is known as the Bar-Shalom/Campo fusion rule [7]. However, cross-correlations between estimation errors are difficult to maintain or to reconstruct. For the purpose of fusing arbitrary estimates, conservative fusion rules, like covariance intersection [11] or ellipsoidal intersection [12], have been proposed, which provide consistent fusion results irrespective of the correlation structure but are often too pessimistic.

This paper is dedicated to a sample-based technique for reconstructing cross-correlations between estimation errors that result from common process noise and the common initialization of the local Kalman filters. In [13], it has been demonstrated for local ensemble Kalman filters that a sample-based reconstruction of cross-correlations is possible. In [14], different random sampling strategies are used to account for correlated and uncorrelated noise terms separately. There, the joint covariance matrix, and in particular the cross-correlations, can be reconstructed from the local sample sets by synchronizing the samplers for the correlated parts. While this joint covariance matrix only asymptotically (in the number of samples) approaches the true one, the technique proposed in this paper allows for an exact reconstruction of the joint covariance matrix. As an important advantage, only a simple deterministic sampling scheme is used to obtain the desired cross-correlations. This approach enables us to optimally fuse locally computed estimates, as explained in the following sections.

II. Problem Formulation and Key Idea

We consider estimating the state $\underline{x}_k$ of a discrete-time stochastic dynamic system based on measurements from several sensor nodes[1]. In order to reduce the amount of measurement data which has to be sent to a fusion center, we pursue distributed state estimation. More precisely, each sensor node performs a local state estimation only by using its own measurements with the Kalman filter or one of its derivatives, such as the unscented Kalman filter (UKF) [15], consisting of the usual alternating prediction steps and measurement updates. Thus, the local state estimate at time step k of the i-th node is a mean vector $\hat{\underline{x}}^{(i)}_{k|k}$ and a covariance matrix $\mathbf{P}^{(i)}_{k|k}$. In doing so, measurements are pre-processed on a node and their information is encoded in the local state estimate. Now, instead of directly sending the measurements to the fusion center, all nodes can simply send their local estimate.

To fuse the local state estimates, the fusion center has to build the joint mean vector

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

and especially the joint covariance matrix

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

where L denotes the number of local estimates to be fused. Based on $\hat{\underline{m}}_{k|k}$ and $\mathbf{J}_{k|k}$, a weighted least squares fusion, i.e., the Bar-Shalom/Campo formulas [7] for the multisensor case, gives the fused state mean and state covariance according to

$$\begin{aligned} \mathbf{P}_{k|k} &= \left(\mathbf{H}^\top \left(\mathbf{J}_{k|k} \right)^{-1} \mathbf{H} \right)^{-1} , \\ \hat{\underline{x}}_{k|k} &= \mathbf{P}_{k|k} \mathbf{H}^\top \left(\mathbf{J}_{k|k} \right)^{-1} \hat{\underline{m}}_{k|k} , \end{aligned} \tag{3}$$

with $\mathbf{H} = \left[\mathbf{I}, \cdots, \mathbf{I} \right]^\top$ and $\mathbf{I}$ the identity matrix of the system state dimension. However, the fusion process requires, in addition to the known covariances $\mathbf{P}^{(i)}_{k|k}$, the cross-covariance matrices $\mathbf{P}^{(i,j)}_{k|k}$ between all local estimates. These are non-zero especially due to the common process noise used in all nodes to model the temporal behavior of the system. Unfortunately, the fusion center cannot reconstruct $\mathbf{P}^{(i,j)}_{k|k}$ only from the local state estimates.

The key idea of our approach to reconstruct the correct cross-covariance matrices $\mathbf{P}^{(i,j)}_{k|k}$ is to use *samples*: besides the actual local state estimate, each node keeps and processes a set of samples that encodes the correlations between its own estimate the state estimates from other nodes. The samples are obtained in a very similar way to the ensemble Kalman filter (EnKF) [16]. Then, in order to compute the correlations, the nodes only have to send their set of samples in addition to their local state estimate to the fusion center. With these, the fusion center can obtain the correlation between the i-th and j-th sensor node by simply computing the sample cross-covariance using their respective sample sets.

The additional overhead of sending the samples generally is cheaper than transmitting the measurements themselves to the fusion center, particularly if each node performs several state predictions and measurement updates between two transmissions. Our sample-based correlation reconstruction approach offers several advantages:

- it can exactly reconstruct the correlations between all local state estimates,
- a sensor node does not need any information about measurements processed at other nodes, neither the used measurement models nor the number of processed measurements,
- the only additional overhead for data transfer are the samples of each node,
- the number of samples required to correctly reconstruct all correlations does not depend on the number of utilized sensor nodes, and thus, does not increase with the number of nodes, and finally
- additional sensor nodes can be simply added over time without affecting the other nodes, i.e., a node is not aware of how many nodes are currently in use for state estimation.

In the next section, we explain the theoretical concepts and workflow of an optimal fusion technique for distributed state estimation using this sample-based correlation reconstruction approach.

III. Optimal Sample-Based Fusion for Distributed Linear State Estimation

After motivating our sample-based correlation reconstruction approach in Section II, we present an optimal sample-based fusion algorithm for linear systems. Based on this, we formulate the sample-based distributed state estimation for nonlinear systems in Section IV.

At its core, our approach relies on a regular data exchange between the sensor nodes and the fusion center. For P time

[1] Vectors are underlined and matrices are printed bold face.

steps, each node performs a local state estimation including the adaption of its correlation samples. After not more than P time steps, nodes send their local state estimate and correlation samples to the fusion center, where this information is used to compute a fused state estimate, which is then sent back to each node. Using this, every node re-initializes its local state estimate and correlation samples with the just fused state estimate, and again locally estimates the system state for P time steps. In this way, the node's local estimation quality profit from the information of the fused estimate. Moreover, the re-initialization of the samples is necessary to guarantee the correctness of the correlations obtained from the samples. In the following, we explain every aspect of our distributed estimation approach step by step, starting with a formal description of the system state estimation.

A. System Description

The temporal evolution of the considered discrete-time stochastic linear dynamic system with state $\underline{x}_k$ is described by

$$\underline{x}_k = \mathbf{A}_k \underline{x}_{k-1} + \mathbf{B}_k \underline{w}_k \ ,$$

where $\underline{w}_k$ denotes zero-mean and white Gaussian noise with covariance $\mathbf{Q}_k$. It is assumed that $\underline{w}_k$ is uncorrelated with the system state for all time steps.

Over time, the i-th sensor node receives noisy measurements $\underline{\tilde{y}}_k^{(i)}$ which are assumed to be generated according to the linear measurement model

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

where $\underline{v}_k^{(i)}$ again denotes zero-mean and white Gaussian noise with covariance $\mathbf{R}_k^{(i)}$. It is also assumed that $\underline{v}_k^{(i)}$ is uncorrelated with the system state, the system noise $\underline{w}_k$, and the measurement noise $\underline{v}_k^{(j)}$ with $j \neq i$ for all time steps.

B. (Re-)Initialization of State Estimate and Correlation Samples

At a regular basis, the fusion center sends the latest fused state estimate $\hat{\underline{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ to all sensor nodes. In particular, for $k = 0$, i.e., the first processing step, $\hat{\underline{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ are sent. Using this, we first re-initialize the local state estimate of the i-th node according to

$$\begin{aligned} \hat{\underline{x}}_{k|k}^{(i)} &= \hat{\underline{x}}_{k|k} \ , \\ \mathbf{P}_{k|k}^{(i)} &= \mathbf{P}_{k|k} \ . \end{aligned} \tag{4}$$

Second, we also need to re-initialize the set of samples used for reconstructing the correlations. To achieve this, we consider the joint space of the system state at time step k and the system noise for the next P time steps, i.e.,

$$\underline{d}_k = \left[\underline{x}_k^\top, \underline{w}_{k+1}^\top, \ldots, \underline{w}_{k+P}^\top\right]^\top \ , \tag{5}$$

where P is to be determined by the user, and controls how many time steps each node can perform before the next node re-initialization is necessary. For an N-dimensional system state $\underline{x}_k$ and a W-dimensional system noise $\underline{w}_k$, the dimension of the joint space $\underline{d}_k$ is $D = N + P \times W$.

Furthermore, all nodes generate the *identical* set $\{\underline{p}^{(m)}\}_{m=1}^M$ of $M = D+1$ equally weighted samples $\underline{p}^{(m)}$ of dimension D, for which the sample mean is zero and the sample covariance matrix yields the identity, that is,

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

The samples are generated with the simple deterministic spherical simplex sampling method proposed in [17]. Note that M cannot be further reduced as $D + 1$ is the minimum number of samples required to represent a valid covariance matrix of dimension $D \times D$.

Then, we construct a covariance matrix for the joint space $\underline{d}_k$ according to

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

The block diagonal structure of $\mathbf{D}_k$ reflects that the system noise is assumed to be white and uncorrelated with the system state for all time steps. By computing the Cholesky decomposition $\mathbf{L}_k \mathbf{L}_k^\top = \mathbf{D}_k$ and using the samples $\{\underline{p}^{(m)}\}_{m=1}^M$, we get a set of M equally weighted samples

$$\begin{aligned} \underline{d}_k^{(m)} &= [(\underline{s}_{k|k}^{(i,m)})^\top, (\underline{w}_{k+1}^{(m)})^\top, \ldots, (\underline{w}_{k+P}^{(m)})^\top]^\top \\ &= \mathbf{L}_k \underline{p}^{(m)} \ , \quad \forall m = 1, \ldots, M \ , \end{aligned} \tag{6}$$

for which again the sample mean is zero, but the sample covariance matrix is equal to $\mathbf{D}_k$. By partitioning the samples $\underline{d}_k^{(m)}$ into the respective subspaces for the system state and for each system noise for the next P time steps, on the one hand we get the desired set $\{\underline{s}_{k|k}^{(i,m)}\}_{m=1}^M$ of the correlation samples for the i-th node, and on the other hand P sets of system noise samples $\{\underline{w}_{k+1}^{(m)}\}_{m=1}^M, \ldots, \{\underline{w}_{k+P}^{(m)}\}_{m=1}^M$. The latter will be used in Section III-C to predict the correlation samples in parallel to the local state estimate.

It is important to note that, although the correlation samples $\{\underline{s}_{k|k}^{(i,m)}\}_{m=1}^M$ lie in the system state space, and thus, the samples are of the state dimension N, they are *not* used to locally estimate the system state. They are only used for encoding the correlations between the local estimates. Moreover, the re-initialized correlation samples and the system noise samples are the *same* on all nodes. This is because the matrix $\mathbf{D}_k$ and samples $\{\underline{p}^{(m)}\}_{m=1}^M$ are identical for all nodes. However, over time, the samples $\{\underline{s}_{k|k}^{(i,m)}\}_{m=1}^M$ will change for each node individually, depending on the performed measurement updates.

As mentioned in Section II, the correlation between node estimates is obtained from the cross-covariance matrix of their respective samples. From the fact that the just re-initialized correlation samples are identical for all nodes, it holds that

$$\begin{aligned} \mathbf{P}_{k|k}^{(i,j)} &= \frac{1}{M} \sum_{m=1}^M \underline{s}_{k|k}^{(i,m)} (\underline{s}_{k|k}^{(j,m)})^\top \\ &= \frac{1}{M} \sum_{m=1}^M \underline{s}_{k|k}^{(i,m)} (\underline{s}_{k|k}^{(i,m)})^\top = \mathbf{P}_{k|k} \ , \quad \forall i \neq j \ . \end{aligned}$$

Consequently, the local state estimates of all nodes are *fully correlated* after a re-initialization. This makes sense, as we re-initialize all nodes with the same system state estimate, and thus, with the same information we have about the current system state.

C. The Time Update

The prediction step on each node is twofold. On the one hand, we have to predict the local state estimate $\underline{\hat{x}}_{k-1|k-1}^{(i)}$ and $\mathbf{P}_{k-1|k-1}^{(i)}$ from time step $k-1$ to k. Due to the linear system model, this is performed with the optimal Kalman filter prediction formulas

$$\begin{aligned} \underline{\hat{x}}_{k|k-1}^{(i)} &= \mathbf{A}_k \underline{\hat{x}}_{k-1|k-1}^{(i)} \ , \\ \mathbf{P}_{k|k-1}^{(i)} &= \mathbf{A}_k \mathbf{P}_{k-1|k-1}^{(i)} \mathbf{A}_k^\top + \mathbf{B}_k \mathbf{Q}_k \mathbf{B}_k^\top \ . \end{aligned} \tag{7}$$

On the other hand, we also have to predict the correlation samples $\{\underline{s}_{k-1|k-1}^{(i,m)}\}_{m=1}^{M}$ from time step $k-1$ to k. This is accomplished by utilizing the corresponding system noise sample set $\{\underline{w}_k^{(m)}\}_{m=1}^{M}$ generated during the node re-initialization according to

$$\underline{s}_{k|k-1}^{(i,m)} = \mathbf{A}_k \underline{s}_{k-1|k-1}^{(i,m)} + \mathbf{B}_k \underline{w}_k^{(m)} \ , \quad \forall m = 1, \ldots, M \ . \tag{8}$$

The sample prediction is identical to a prediction done by the EnKF, except for the selection of system noise samples. In the EnKF, usually a random sample is drawn from the system noise distribution for each state sample. Such noise samples, however, do neither necessarily reflect the correct distribution of the system noise nor guarantee noise samples which are uncorrelated with the state samples or with the noise samples from past prediction steps.

Finally, remember that the used noise samples $\{\underline{w}_k^{(m)}\}_{m=1}^{M}$ are the same on all nodes. This fact reflects the common process noise all nodes have to take into account.

D. The Measurement Update

If a new measurement $\underline{\tilde{y}}_k^{(i)}$ is available at time step k, we can perform a measurement update on the i-th node. Compared to the state prediction, the measurement update of the local state estimate and the correlation samples are more coupled. First, we compute the Kalman gain

$$\mathbf{K}_k^{(i)} = \mathbf{P}_{k|k-1}^{(i)} (\mathbf{H}_k^{(i)})^\top \left(\mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} (\mathbf{H}_k^{(i)})^\top + \mathbf{R}_k^{(i)} \right)^{-1} \ .$$

Based on this, we perform the common Kalman filter update for the local state estimate

$$\begin{aligned} \underline{\hat{x}}_{k|k}^{(i)} &= \underline{\hat{x}}_{k|k-1}^{(i)} + \mathbf{K}_k^{(i)} (\underline{\tilde{y}}_k^{(i)} - \mathbf{H}_k^{(i)} \underline{\hat{x}}_{k|k-1}^{(i)}) \ , \\ \mathbf{P}_{k|k}^{(i)} &= (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \mathbf{P}_{k|k-1}^{(i)} \ . \end{aligned} \tag{9}$$

Finally, we also use the Kalman gain to update each correlation sample individually

$$\underline{s}_{k|k}^{(i,m)} = (\mathbf{I} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)}) \underline{s}_{k|k-1}^{(i,m)} \ , \quad \forall m = 1, \ldots, M \ . \tag{10}$$

Note that, in contrast to the time update, we do not need any noise samples from the measurement noise distribution. In addition, the measurement $\underline{\tilde{y}}_k^{(i)}$ also has no influence on the correlation samples. This is different from the EnKF measurement update, where noise samples are strictly necessary and also the measurement influences the state samples. However, due to the fact that we only use the samples to obtain correlations (not the local state estimate itself) and that the measurement noise from different nodes are assumed to be uncorrelated, measurement and noise samples are omitted in our case.

In case of no measurement being available at sensor node i, we simply keep the predicted quantities, i.e., $\underline{\hat{x}}_{k|k}^{(i)} = \underline{\hat{x}}_{k|k-1}^{(i)}$, $\mathbf{P}_{k|k}^{(i)} = \mathbf{P}_{k|k-1}^{(i)}$, and $\underline{s}_{k|k}^{(i,m)} = \underline{s}_{k|k-1}^{(i,m)} \ , \ \forall m = 1, \ldots, M$.

E. The Optimal Fusion

After all nodes have performed P prediction steps (and, potentially, measurement updates in between) since the last re-initialization at time step k, we want to compute an optimal fused state estimate at the fusion center. For this purpose, each node sends their current local estimate $\underline{\hat{x}}_{k+P|k+P}^{(i)}$ and $\mathbf{P}_{k+P|k+P}^{(i)}$ together with its set of correlation samples $\{\underline{s}_{k+P|k+P}^{(i,m)}\}_{m=1}^{M}$ to the fusion center.

Here, the joint state mean (1) is built out of the local state means $\underline{\hat{x}}_{k+P|k+P}^{(i)}$. Additionally, we compute the correlations between all local state estimates according to

$$\mathbf{P}_{k+P|k+P}^{(i,j)} = \frac{1}{M} \sum_{m=1}^{M} \underline{s}_{k+P|k+P}^{(i,m)} (\underline{s}_{k+P|k+P}^{(j,m)})^\top \ , \tag{11}$$

with $i \neq j$. The proof of the correctness of the correlations obtained from (11) is given in the Appendix. Together with the local state covariances $\mathbf{P}_{k+P|k+P}^{(i)}$, the joint state covariance matrix (2) is built. Finally, with the aid of (3), the fused state mean $\underline{\hat{x}}_{k+P|k+P}$ and the fused state covariance matrix $\mathbf{P}_{k+P|k+P}$ is obtained. This fused state estimate is then transferred back to each node to re-initialize their local estimates and correlation samples.

Note that we only have to fuse the local state estimates of those nodes, which have conducted at least one measurement update since the last re-initialization at time step k. Nodes, which have only performed prediction steps, yield no additional information about the current system state, and thus, their estimates can be ignored. Please also note that the nodes do not necessarily have to perform P predictions before a fusion can be accomplished.

F. Summary

Algorithm 1 summarizes the entire sample-based distributed state estimation procedure in the case of S sensor nodes. Additionally, Figure 1 illustrates the data flow in the sensor network. It can be seen from the data flow that a node does not need to know any information about measurements processed at other nodes, neither the used measurement models nor the number of processed measurements. Moreover, additional nodes can be added over time by simply sending the fused state mean and state covariance matrix at the next node re-initialization.

The maximum number of prediction steps P can be defined by the user. The choice depends on how often a fused state

Algorithm 1 Sample-Based Distributed State Estimation

```
Set initial state estimate $\hat{\underline{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ at the fusion center
for $k = 0, P, 2P, \ldots$ do
    Fusion center sends current state estimate $\hat{\underline{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ to all nodes
    for $1 \leq i \leq S$ do
        $i$-th node sets local state estimate according to (4)
        $i$-th node gets correlation samples according to (6)
        for $1, \ldots, P$ do
            $i$-th node performs state prediction (7), (8)
            $i$-th node performs measurement update (9), (10)
        end for
        $i$-th node sends estimate $\hat{\underline{x}}^{(i)}_{k+P|k+P}$ and $\mathbf{P}^{(i)}_{k+P|k+P}$, and samples $\{\underline{s}^{(i,m)}_{k+P|k+P}\}^{M}_{m=1}$ to fusion center
    end for
    Fusion center computes new state estimate $\hat{\underline{x}}_{k+P|k+P}$ and $\mathbf{P}_{k+P|k+P}$ with (1)–(3) and (11)
end for
```

estimate is required and how much data for the correlation samples is allowed to be sent. The larger P the larger will be the joint space of system state and system noise (5). Thus, more correlation samples are required, which have to be sent to the fusion center. However, keep in mind that the memory consumption only for the correlation samples is

$$N \times M = N^2 + N(P \times W) + N \ .$$

As a consequence, the data that has to be sent to the fusion center increases only linearly in the maximum number of prediction steps P.

Furthermore, as each node can generate the necessary samples $\{\underline{p}^{(m)}\}^{M}_{m=1}$ on their own, they *never* have to be sent through the network. Note also that the order of correlation samples and system noise samples is important, and has to be the same for all nodes to correctly compute the correlations.

Instead of using (6) to obtain the system noise samples $\{\underline{w}^{(m)}_k\}^{M}_{m=1}$, we could rely on pseudo random number generators with the same seed for each node. However, the resulting noise samples would only asymptotically approach the assumed whiteness and uncorrelatedness with the system state. Thus, in practice the fused state estimate would never be identical to the optimal solutions like our approach. Moreover, the only way to mitigate the errors introduced by the violated correlation assumptions is to use many random samples, which in turn would drastically increase the communication overhead.

IV. Sample-Based Fusion for Distributed State Estimation in the Nonlinear Case

In this section, we apply the sample-based distributed state estimation technique from Section III to nonlinear problems. More precisely, for the i-th sensor node we now consider the nonlinear measurement model

$$\underline{y}^{(i)}_k = \underline{h}^{(i)}_k(\underline{x}_k) + \underline{v}^{(i)}_k \ , \tag{12}$$

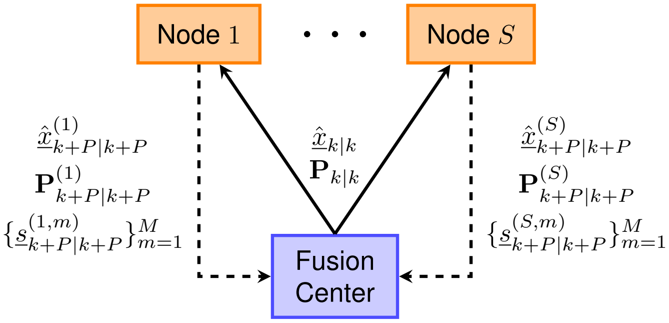

Node 1
Node S
$\hat{\underline{x}}^{(1)}_{k+P|k+P}$
$\mathbf{P}^{(1)}_{k+P|k+P}$
$\{\underline{s}^{(1,m)}_{k+P|k+P}\}^{M}_{m=1}$
$\hat{\underline{x}}_{k|k}$
$\mathbf{P}_{k|k}$
$\hat{\underline{x}}^{(S)}_{k+P|k+P}$
$\mathbf{P}^{(S)}_{k+P|k+P}$
$\{\underline{s}^{(S,m)}_{k+P|k+P}\}^{M}_{m=1}$
Fusion Center

Figure 1: Data flow over time in the sample-based distributed state estimation approach. Every P time steps, the nodes send their estimates and samples to the fusion center, where this information is used to compute a fused state estimate, which is then sent back to each node.

where $\underline{v}^{(i)}_k$ is defined as in Section III.

In order to apply (12) to our distributed estimation algorithm, we need linear approximations. These approximations can be obtained, for example, using explicit linearization around the prior state mean as it is done by the extended Kalman filter (EKF) [18]. However, this type of linearization is very sensitive to the prior state mean, as it does not consider the state uncertainty for the linearization. In particular, an EKF using difference quotient approximations for the Jacobian did not work out in our target tracking evaluation presented in Section V. That is, the estimation results from our approach diverged significantly from the optimal solutions.

Instead, we propose to rely on statistical linearization, e.g., done by sample-based Kalman filters such as the unscented Kalman filter (UKF) [15] or the smart sampling Kalman filter (S^{2}KF) [19], [20]. Statistical linearization requires to compute the measurement mean

$$\hat{\underline{y}}^{(i)}_k = \int_{\mathbb{R}^N} \underline{h}^{(i)}_k(\underline{x}_k)\mathcal{N}(\underline{x}_k; \hat{\underline{x}}^{(i)}_{k|k-1}, \mathbf{P}^{(i)}_{k|k-1}) \ \mathrm{d}\underline{x}_k \ ,$$

the measurement covariance matrix

$$\begin{aligned}\mathbf{Y}^{(i)}_k = \int_{\mathbb{R}^N} &(\underline{h}^{(i)}_k(\underline{x}_k) - \hat{\underline{y}}^{(i)}_k)(\underline{h}^{(i)}_k(\underline{x}_k) - \hat{\underline{y}}^{(i)}_k)^{\top} \cdot \\ &\mathcal{N}(\underline{x}_k; \hat{\underline{x}}^{(i)}_{k|k-1}, \mathbf{P}^{(i)}_{k|k-1}) \, \mathrm{d}\underline{x}_k + \mathbf{R}^{(i)}_k \ ,\end{aligned}$$

and the cross-covariance matrix of predicted state and measurement

$$\begin{aligned}\mathbf{C}^{(i)}_k = \int_{\mathbb{R}^N} &(\underline{x}_k - \hat{\underline{x}}^{(i)}_{k|k-1})(\underline{h}^{(i)}_k(\underline{x}_k) - \hat{\underline{y}}^{(i)}_k)^{\top} \cdot \\ &\mathcal{N}(\underline{x}_k; \hat{\underline{x}}^{(i)}_{k|k-1}, \mathbf{P}^{(i)}_{k|k-1}) \, \mathrm{d}\underline{x}_k \ .\end{aligned}$$

In general, these integrals cannot be solved analytically, and approximations from sample-based Kalman filters are the means of choice to obtain them.

Using the above moments, we can get an optimal linear approximation of (12) in the Mean Square Error (MSE) sense, e.g., see [21]. Based on this linear approximation, we can

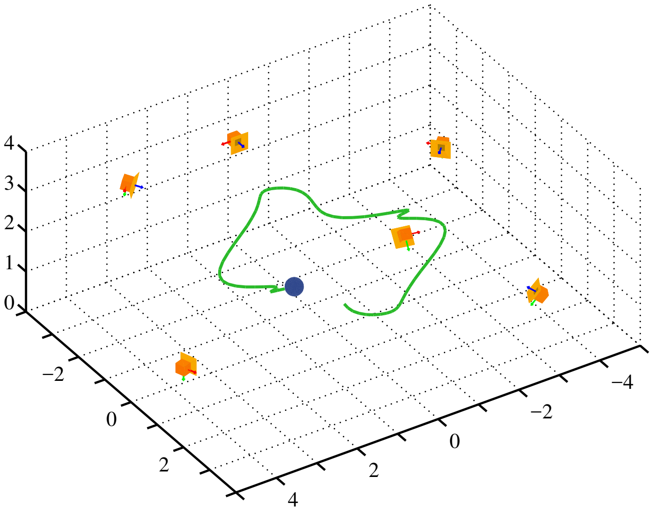


Figure 2: Sensor network consisting of six Microsoft Kinect cameras (orange), sphere to be tracked (blue), and parts of the sphere's trajectory (green).

change (9) and (10) according to

$$\begin{aligned} \hat{\underline{x}}_{k|k}^{(i)} &= \hat{\underline{x}}_{k|k-1}^{(i)} + \mathbf{C}_k^{(i)}(\mathbf{Y}_k^{(i)})^{-1}(\tilde{\underline{y}}_k^{(i)} - \hat{\underline{y}}_k^{(i)}) \ , \\ \mathbf{P}_{k|k}^{(i)} &= (\mathbf{I} - \mathbf{C}_k^{(i)}(\mathbf{Y}_k^{(i)})^{-1}(\mathbf{C}_k^{(i)})^\top(\mathbf{P}_{k|k-1}^{(i)})^{-1})\mathbf{P}_{k|k-1}^{(i)} \ , \\ \underline{s}_{k|k}^{(i,m)} &= (\mathbf{I} - \mathbf{C}_k^{(i)}(\mathbf{Y}_k^{(i)})^{-1}(\mathbf{C}_k^{(i)})^\top(\mathbf{P}_{k|k-1}^{(i)})^{-1})\underline{s}_{k|k-1}^{(i,m)} \end{aligned}$$

in order to perform the (linear) measurement update from Section III-D for the nonlinear model (12).

V. Evaluation

In this section, we show that our sample-based correlation reconstruction approach yields the same results as the optimal fusion where the exact correlations are at hand. Our goal is to track a sphere of known radius $r = 10\,\mathrm{cm}$ in a network consisting of six Microsoft Kinect depth cameras (see Figure 2) like in [22]. Each camera is a separate node that only processes its own measurements. The number of time steps between two estimate fusion operations is set to $P = 5$. In addition, the first Kinect node acts as fusion center. This implies that in each time step, where no fusion of local state estimates happens, only the local state estimate of this node will serve as the "fused" state estimate.

The to be estimated system state $\underline{x}_k = [\underline{c}_k^\top, \underline{\nu}_k^\top]^\top$ comprises the sphere's position $\underline{c}_k = [c_k^x, c_k^y, c_k^z]^\top$ as well as its velocity $\underline{\nu}_k = [\nu_k^x, \nu_k^y, \nu_k^z]^\top$. We model the temporal evolution of the sphere by means of a constant velocity system model with time-invariant matrices

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

where the time period T is $30\,\mathrm{ms}$.

Each Kinect camera provides us with a large number of 3D noisy point measurements per time step from the sphere's surface. Additionally, a camera can only generate measurements within a limited field of view, and as a consequence, a camera does not measure points in each time step. For the measurement model, we assume that each point measurement $\tilde{\underline{y}}_k^{(i)}$ is corrupted by additive white Gaussian noise $\underline{v}_k^{(i)}$ with covariance $\mathbf{R}_k^{(i)}$, which leads to the measurement model

$$\underline{y}_k^{(i)} = \underline{z}_k^{(i)} + \underline{v}_k^{(i)} \ ,$$

where $\underline{z}_k^{(i)}$ denotes the point on the sphere's surface from which the measurement $\tilde{\underline{y}}_k^{(i)}$ originates, called the *measurement source*. $\mathbf{R}_k^{(i)}$ is given by the non-isotropic Kinect measurement noise proposed in [23].

Algorithm 2 computeSource (see [22])

```
Input: State $\underline{x}_k$, camera position $\underline{m}$, and measurement $\tilde{\underline{y}}_k^{(i)}$
  $\underline{d}$ = normalize$(\tilde{\underline{y}}_k^{(i)} - \underline{m})$    $\underline{e} = \tilde{\underline{y}}_k^{(i)} - \underline{c}_k$
  $a = \underline{d}^\top \underline{e}$    $b = a^2 - (\underline{e}^\top \underline{e} - r^2)$
  if $b < 0$ then
    $\underline{n}$ = normalize$(\underline{c}_k - \underline{m})$
    if $\underline{e}^\top \underline{n} < 0$ then
      $\underline{z}_k^{(i)} = \underline{c}_k + r\,$normalize$(\underline{e})$
    else
      $\underline{l} = \tilde{\underline{y}}_k^{(i)} - (\underline{e}^\top \underline{n} / \underline{d}^\top \underline{n})\underline{d}$
      $\underline{z}_k^{(i)} = \underline{c}_k + r\,$normalize$(\underline{l} - \underline{c}_k)$
    end if
  else
    $d_1 = -a + \sqrt{b}$    $d_2 = -a - \sqrt{b}$
    if $d_1 < d_2$ then
      $\underline{z}_k^{(i)} = \tilde{\underline{y}}_k^{(i)} + d_1\underline{d}$
    else
      $\underline{z}_k^{(i)} = \tilde{\underline{y}}_k^{(i)} + d_2\underline{d}$
    end if
  end if
Output: Measurement source $\underline{z}_k^{(i)}$
```

Unfortunately, for each measurement $\tilde{\underline{y}}_k^{(i)}$, the true measurement source $\underline{z}_k^{(i)}$ is not known. However, in order to obtain the most probable source $\underline{z}_k^{(i)}$ for each measurement, we make use of the so-called Greedy Association Model (GAM) [24]. To further improve the state estimation quality, we also make use of the known position $\underline{m}$ of a Kinect camera to exploit the geometrical interaction between camera, measurement, and sphere. This leads to the nonlinear measurement source approximation

$$\underline{z}_k^{(i)} \approx \mathrm{computeSource}(\underline{x}_k, \underline{m}, \tilde{\underline{y}}_k^{(i)}) \ ,$$

listed in Algorithm 2. A more detailed description of this measurement model can be found in [22]. In order to perform the measurement updates, we use the moment approximations from the UKF to compute the required moments $\hat{\underline{y}}_k^{(i)}$, $\mathbf{Y}_k^{(i)}$, and $\mathbf{C}_k^{(i)}$. Furthermore, we process measurements from a single time step sequentially on each node.

We perform 50 Monte Carlo runs. In each run, the sphere moves along the same nonlinear trajectory and we simulate appropriate noisy measurements for the six Kinect sensors. Thus, the number of measurements (see Figure 3a) does not change between the runs. We compare our proposed sample-based fusion approach against the optimal fusion where the exact correlations are at hand. That is, we also use the UKF to get the linear approximations for (12), but compute the exact correlations with the knowledge of the measurements from all nodes.

The results of the sphere position and sphere velocity Root Mean Square Errors (RMSEs) are plotted in Figures 3b and 3c, respectively. It can be seen that the errors are the smallest in those time steps, when the sphere is visible for the fusion center node (“Kinect 1” in Figure 3a). In the other cases, the respective errors have a sawtooth-shaped curve. This is due to the constant velocity model and that fusions only happen every fifth time step. But more importantly, the error curves of the sample-based fusion and optimal fusion are identical for all time steps. More precisely, the fused state means and state covariance matrices only differ in an order of magnitude of $\approx 10^{-12}$. This small difference can be explained by numerical effects imposed by the sample-based approach.

In our setup, each node requires $M = 22$ samples ($D = 6 + 5 \times 3 = 21$) for the correlation reconstruction. In total, this makes only $6 \times 22 = 132$ values to be sent every fifth time step from each node to the fusion center (besides the actual local state estimate, of course). Opposed to this, on each node, thousands of 3D measurements are available per time step. Sending these would be much more demanding.

VI. Conclusions

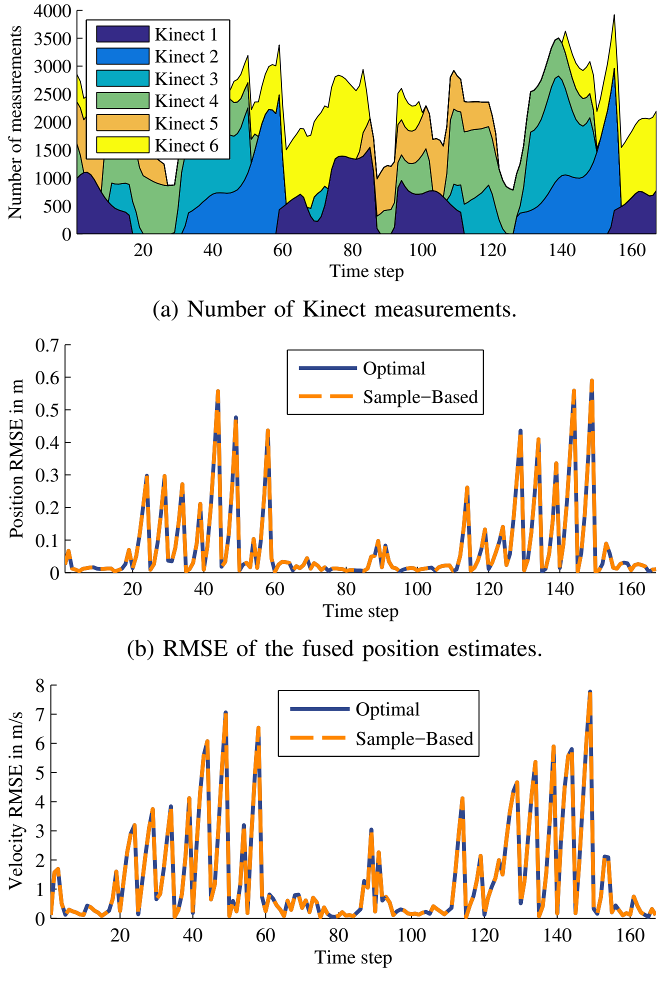
In this paper, we proposed an optimal sample-based fusion technique for distributed state estimation. The key idea of our approach is to keep and process a small number of deterministic samples on each sensor node. In order to correctly reconstruct the correlations between local system state estimates, the nodes only have to send their respective samples to the fusion center, where the desired correlations are simply obtained by computing sample cross-covariance matrices. In doing so, a sensor node does not need any information about measurements processed at other nodes. Moreover, the approach scales very well with the number of utilized nodes. As expected, the nonlinear target tracking evaluation showed that our sample-based fusion yields the same results as the optimal fusion.

Appendix

We prove that the correlation between two sensor nodes i and j can be correctly reconstructed when their respective set of samples is initialized with (6), predicted with (8), and updated according to (10).

It is assumed that both nodes are re-initialized in time step k, and that the local state estimates are fully correlated with $\mathbf{P}_{k|k}^{(i,j)} = \mathbf{P}_{k|k}$. Moreover, w.l.o.g. it is assumed that

1) the i-th node performs a prediction from time step k to $k+1$ followed by an update in time step $k+1$ and another prediction from time step $k+1$ to $k+2$, and
2) the j-th node performs a prediction from time step k to $k+1$ followed by another prediction from time step $k+1$ to $k+2$ and finally performs an update in time step $k+2$.



(a) Number of Kinect measurements.

(b) RMSE of the fused position estimates.

(c) RMSE of the fused velocity estimates.

Figure 3: Distributed sphere tracking estimation results.

Then, the correct correlation between their local estimates at time step $k+2$ is [25]

$$\mathbf{P}_{k+2|k+2}^{(i,j)} = (\mathbf{A}_{k+2}\mathbf{D}_3(\mathbf{D}_1 + \mathbf{D}_2)\mathbf{A}_{k+2}^\top + \mathbf{D}_4)\mathbf{D}_5^\top \ ,$$

with

$$\begin{aligned}
\mathbf{D}_1 &= \mathbf{A}_{k+1}\mathbf{P}_{k|k}^{(i,j)}\mathbf{A}_{k+1}^\top = \mathbf{A}_{k+1}\mathbf{P}_{k|k}\mathbf{A}_{k+1}^\top \ , \\
\mathbf{D}_2 &= \mathbf{B}_{k+1}\mathbf{Q}_{k+1}\mathbf{B}_{k+1}^\top \ , \quad \mathbf{D}_3 = \mathbf{I} - \mathbf{K}_{k+1}^{(i)}\mathbf{H}_{k+1}^{(i)} \ , \\
\mathbf{D}_4 &= \mathbf{B}_{k+2}\mathbf{Q}_{k+2}\mathbf{B}_{k+2}^\top \ , \quad \mathbf{D}_5 = \mathbf{I} - \mathbf{K}_{k+2}^{(j)}\mathbf{H}_{k+2}^{(j)} \ .
\end{aligned}$$

The correlation samples of the i-th node are

$$\begin{aligned}
\underline{s}_{k+2|k+2}^{(i,m)} = {} & \mathbf{B}_{k+2}\underline{w}_{k+2}^{(m)} + \mathbf{A}_{k+2}\mathbf{D}_3\mathbf{B}_{k+1}\underline{w}_{k+1}^{(m)} \\
& + \mathbf{A}_{k+2}\mathbf{D}_3\mathbf{A}_{k+1}\underline{s}_{k|k}^{(i,m)} \ , \quad \forall m = 1, \ldots, M \ ,
\end{aligned}$$

and the correlation samples of the j-th node are

$$
\begin{aligned}
(\underline{s}^{(j,m)}_{k+2|k+2})^\top &= (\underline{w}^{(m)}_{k+2})^\top \mathbf{B}^\top_{k+2}\mathbf{D}^\top_5 + (\underline{w}^{(m)}_{k+1})^\top \mathbf{B}^\top_{k+1}\mathbf{A}^\top_{k+2}\mathbf{D}^\top_5 \\
&\quad + (\underline{s}^{(j,m)}_{k|k})^\top \mathbf{A}^\top_{k+1}\mathbf{A}^\top_{k+2}\mathbf{D}^\top_5 \ , \ \forall m = 1, \ldots, M \ .
\end{aligned}
$$

First, from the fact that the samples obtained from (6) have zero mean, we can see that both sample means are zero according to

$$
\begin{aligned}
\hat{\underline{s}}^{(i)}_{k+2|k+2} &= \mathbf{A}_{k+2}\mathbf{D}_3\mathbf{A}_{k+1}\hat{\underline{s}}^{(i)}_{k|k} = \underline{0} \ , \\
\hat{\underline{s}}^{(j)}_{k+2|k+2} &= \mathbf{D}_5\mathbf{A}_{k+2}\mathbf{A}_{k+1}\hat{\underline{s}}^{(j)}_{k|k} = \underline{0} \ .
\end{aligned}
$$

Second, for both nodes the samples obtained from (6) are identical, and hence, it holds that

$$
\begin{aligned}
\frac{1}{M}\sum_{m=1}^{M} \underline{s}^{(i,m)}_{k|k}(\underline{s}^{(j,m)}_{k|k})^\top &= \mathbf{P}_{k|k} \ , \\
\frac{1}{M}\sum_{m=1}^{M} \underline{w}^{(m)}_{k+1}(\underline{w}^{(m)}_{k+1})^\top &= \mathbf{Q}_{k+1} \ , \\
\frac{1}{M}\sum_{m=1}^{M} \underline{w}^{(m)}_{k+2}(\underline{w}^{(m)}_{k+2})^\top &= \mathbf{Q}_{k+2} \ ,
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbf{0} &= \frac{1}{M}\sum_{m=1}^{M} \underline{s}^{(i,m)}_{k|k}(\underline{w}^{(m)}_{k+1})^\top = \frac{1}{M}\sum_{m=1}^{M} \underline{s}^{(i,m)}_{k|k}(\underline{w}^{(m)}_{k+2})^\top \\
&= \frac{1}{M}\sum_{m=1}^{M} \underline{s}^{(j,m)}_{k|k}(\underline{w}^{(m)}_{k+1})^\top = \frac{1}{M}\sum_{m=1}^{M} \underline{s}^{(j,m)}_{k|k}(\underline{w}^{(m)}_{k+2})^\top \\
&= \frac{1}{M}\sum_{m=1}^{M} \underline{w}^{(m)}_{k+1}(\underline{w}^{(m)}_{k+2})^\top \ .
\end{aligned}
$$

Finally, by exploiting these, the correlation between both nodes obtained from their respective samples is

$$
\begin{aligned}
\mathbf{S}^{(i,j)}_{k+2|k+2} &= \frac{1}{M}\sum_{m=1}^{M} \underline{s}^{(i,m)}_{k+2|k+2}(\underline{s}^{(j,m)}_{k+2|k+2})^\top \\
&= \mathbf{A}_{k+2}\mathbf{D}_3\mathbf{A}_{k+1}\mathbf{P}_{k|k}\mathbf{A}^\top_{k+1}\mathbf{A}^\top_{k+2}\mathbf{D}^\top_5 \\
&\quad + \mathbf{A}_{k+2}\mathbf{D}_3\mathbf{B}_{k+1}\mathbf{Q}_{k+1}\mathbf{B}^\top_{k+1}\mathbf{A}^\top_{k+2}\mathbf{D}^\top_5 \\
&\quad + \mathbf{B}_{k+2}\mathbf{Q}_{k+2}\mathbf{B}^\top_{k+2}\mathbf{D}^\top_5 \\
&= (\mathbf{A}_{k+2}\mathbf{D}_3(\mathbf{D}_1 + \mathbf{D}_2)\mathbf{A}^\top_{k+2} + \mathbf{D}_4)\mathbf{D}^\top_5 \\
&= \mathbf{P}^{(i,j)}_{k+2|k+2} \ .
\end{aligned}
$$

References

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

[2] Kuo-Chu Chang, Rajat K. Saha, and Yaakov Bar-Shalom, "On Optimal Track-to-Track Fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, Oct. 1997.

[3] Hamid R. Hashemipour, Sumit Roy, and Alan J. Laub, "Decentralized Structures for Parallel Kalman Filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, Jan. 1988.

[4] Felix Govaers and Wolfgang Koch, "On the Globalized Likelihood Function for Exact Track-To-Track Fusion at Arbitrary Instants of Time," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[5] Marc Reinhardt, Benjamin Noack, and Uwe 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.

[6] Marc Reinhardt, Benjamin Noack, and Uwe D. Hanebeck, "Advances in Hypothesizing Distributed Kalman Filtering," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[7] Yaakov Bar-Shalom and Leon Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, no. 6, pp. 803–805, Nov. 1986.

[8] Chee-Yee Chong, Kuo-Chu Chang, and Shozo Mori, "Distributed tracking in distributed sensor networks," in *Proceedings of the 1986 American Control Conference (ACC 1986)*, Seattle, Washington, USA, 1986.

[9] Chee-Yee Chong, Shozo Mori, Felix Govaers, and Wolfgang Koch, "Comparison of Tracklet Fusion and Distributed Kalman Filter for Track Fusion," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[10] Wolfgang Koch and Felix Govaers, "On Accumulated State Densities with Applications to Out-of-Sequence Measurement Processing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2766–2778, Oct. 2011.

[11] Simon J. Julier and Jeffrey 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.

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

[13] Jan Čurn, Dan Marinescu, Niall O'Hara, and Vinny 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.

[14] Marc Reinhardt, Benjamin Noack, and Uwe 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.

[15] Simon J. Julier and Jeffrey K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," in *Proceedings of the IEEE*, vol. 92, Mar. 2004, pp. 401–422.

[16] Geir Evensen, "Sequential Data Assimilation with a Nonlinear Quasi-Geostrophic Model Using Monte Carlo Methods to Forecast Error Statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, May 1994.

[17] Simon J. Julier, "The Spherical Simplex Unscented Transformation," in *Proceedings of the 2003 American Control Conference (ACC 2003)*, Jun. 2003, pp. 2430–2434.

[18] Dan Simon, *Optimal State Estimation*, 1st ed. Wiley & Sons, 2006.

[19] Jannik Steinbring and Uwe D. Hanebeck, "LRKF Revisited: The Smart Sampling Kalman Filter (S^{2}KF)," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, Dec. 2014.

[20] Jannik Steinbring, "Nonlinear Estimation Toolbox." [Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox

[21] Ángel F. García-Fernández, Lennart Svensson, Mark Morelande, and Simo Särkkä, "Posterior Linearisation Filter: Principles and Implementation Using Sigma Points," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5561–5573, Oct. 2015.

[22] Jannik Steinbring and Uwe D. Hanebeck, "GPU-Accelerated Progressive Gaussian Filtering with Applications to Extended Object Tracking," in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015, pp. 1038–1045.

[23] Florian Faion, Marcus Baum, and Uwe D. Hanebeck, "Tracking 3D Shapes in Noisy Point Clouds with Random Hypersurface Models," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[24] Florian Faion, Antonio Zea, and Uwe D. Hanebeck, "Reducing Bias in Bayesian Shape Estimation," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[25] Shozo Mori, Kuo-Chu Chang, and Chee-Yee Chong, *Distributed Data Fusion for Network-Centric Operations*, ser. The Electrical Engineering and Applied Signal Processing Series. CRC Press, 2013, ch. Essence of Distributed Target Tracking, pp. 125–160.