Nonlinear Federated Filtering

Benjamin Noack*, **Simon J. Julier****, **Marc Reinhardt***, and **Uwe D. Hanebeck***

*Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
{benjamin.noack,marc.reinhardt,uwe.hanebeck}@ieee.org

**Virtual Environments and Computer Graphics Group (VECG)
Department of Computer Science
University College London (UCL), United Kingdom
s.julier@ucl.ac.uk

***Abstract*—The federated Kalman filter embodies an efficient and easy-to-implement solution for linear distributed estimation problems. Data from independent sensors can be processed locally and in parallel on different nodes without running the risk of erroneously ignoring possible dependencies. The underlying idea is to counteract the common process noise issue by inflating the joint process noise matrix. In this paper, the same trick is generalized to nonlinear models and non-Gaussian process noise. The probability density of the joint process noise is split into an exponential mixture of transition densities. By this means, the process noise is modeled to independently affect the local system models. The estimation results provided by the sensor devices can then be fused, just as if they were indeed independent.**

***Index Terms*—Federated Kalman Filter, Nonlinear Estimation, Distributed Estimation.**

I. Introduction

Distributed state estimation algorithms [1], [2] are of key importance for many network-based applications. The estimation system may be compiled of multiple sensor devices and their output has to converted into relevant information. Proposed advantages include scalability, robustness to failures, and sometimes simply the need to use networked systems, e.g. in order to monitor a large-scale phenomenon, that call for distributed state estimation algorithms. The optimal solution is a centralized architecture that significantly eases the design of state estimation methods, but high measurement frequencies then may exhaust communication bandwidth and jeopardize the node's power resources. Therefore, one strategy is to hand over the processing tasks to the individual sensing nodes, and they rather deliver estimates than sensor readings. Although sensor networks are usually characterized by the idea to process and collect data locally and independently on sensor nodes, this does by no means mean that the data are independent of each other. In order to assemble the big picture of the considered phenomenon, the locally processed estimates have to be transmitted to and fused at a data sink. A lack of independence often entails considerably more elaborate fusion methods so as to avoid erroneous fusion results.

Dependencies among local estimates generally can be traced back to common prior/sensor information and common process noise. In particular, the former source of dependency is encountered in fully decentralized networks, where each node updates its local estimate by data exchange with neighboring nodes. Cycles within a communication path may occur, and hence data may simply be double-counted. As a countermeasure, common information between two sensor nodes can be stored somewhere and be removed from the fusion result, but this requires certain non-decentralized network topologies. Especially in hierarchical networks, the information filter [1], [3] can be utilized to keep track of common information. The simplest solution is a single data sink, which is the only node that fuses local estimates. However, the problem of common process noise [4] —the second cause of dependencies—cannot be resolved this way. The same process noise is modeled on several sensor nodes, which implies full correlation between the local noise terms. Known cross-correlations between local estimates can be exploited by the Bar-Shalom/Campo fusion rule [4], which guarantees minimum mean squared error (MMSE) fusion[1] results. However, bookkeeping of cross-correlations is often far too expensive or even impossible for a myriad of nodes. Parallelized or distributed versions [6], [7] of the Kalman filter algorithm [8] can be employed to circumvent the problem of common process noise. These approaches provide MMSE estimates but suffer from strict requirements regarding the local availability of knowledge about the utilized models, i.e., each node must be aware of the other nodes' sensor models, noise characteristics, and measurement frequencies. Furthermore, the local estimates rather represent meaningless variables than valid estimates and must be available in their entirety at the fusion center in order to obtain a valid state estimate. In particular, the failure of a single component may bring down the entire state estimation system. These properties may limit or even prevent scalability and robustness.

An alternative direction is to move away from optimal towards suboptimal estimation and fusion methods. A well-known example is the covariance intersection (CI) algorithm [9], [10], which provides covariance-consistent fusion results, irrespective of the actual cross-correlations. Consequently, this algorithm is eminently suitable for use in fully decentralized sensor networks. However, compared to centralized methods, the provided fusion results can be too conservative. In [11], the distributed Kalman filter [7] has been improved to degrade gracefully depending on how well the assumptions on the global network are met, i.e., to which extent the nodes are aware of each other. It can be seen as a basic principle that the more knowledge nodes

[1]The MMSE fusion of local Kalman filter results does not resemble an MMSE estimator [5], i.e., a central Kalman filter may provide better results than local filters with subsequent fusion.

have about the entire network the closer to optimal the fusion results can be. The federated Kalman filter (FKF) [12]–[14] can be regarded as a predecessor of the optimal distributed Kalman filter [7] and, as such, it requires less assumptions to be met. The only knowledge the sensor nodes need is the number of participating sensor nodes so as to "decorrelate" the joint process noise matrix. In this way, the nodes can operate independently, and their estimation results can be fused at the data sink as if they were independent. Compared to CI, the FKF requires a data sink for fusion but inherits the advantage that it provides less conservative fusion results since only the common process noise is bounded.

Although linear distributed and decentralized estimation problems still deserve attention, the question arises as to how to generalize the aforementioned approaches to nonlinear Bayesian estimation problems. The most important contribution to this question is the exponential mixture density (EMD) fusion rule [15]–[19], which is a consequent generalization of the CI algorithm and its usefulness has been underlined, for instance, against the background of distributed tracking scenarios [20], [21]. The EMD rule is always applicable to fuse estimates but, like its linear counterpart, can provide very conservative results. Again, less conservative estimates are achievable if additional knowledge can be exploited. For networks where only the treatment and removal of common information [22] is required, nonlinear information filters can be derived [23]. In order to complete the picture, also a nonlinear counterpart to the federated Kalman filter is desirable that is tailor-made to counteract the problem of common process noise. In many applications, local estimates are solely fused in a data sink, and hence only common process noise causes dependencies. The use of a nonlinear version of the FKF, as derived in this paper, can then significantly contribute to reducing conservatism.

PRELIMINARIES

Underlined variables $\underline{x}$ denote vectors or vector-valued functions, and lowercase boldface letters $\underline{\boldsymbol{x}}$ are used for random quantities. Matrices are written in uppercase boldface letters $\mathbf{C}$. By $(\underline{\hat{x}}, \mathbf{C})$, we denote an estimate with mean $\underline{\hat{x}}$ and covariance matrix $\mathbf{C}$. The function $\mathcal{N}(\cdot; \underline{\hat{x}}, \mathbf{C})$ is the Gaussian density

$$f(\cdot) = C^{-1} \cdot \exp\left\{-\tfrac{1}{2}\big(\cdot - \underline{\hat{x}}\big)^{\mathrm{T}} (\mathbf{C})^{-1} \big(\cdot - \underline{\hat{x}}\big)\right\} \ .$$

with mean $\underline{\hat{x}}$ and covariance matrix $\mathbf{C}$. The scalar C is the normalization factor. The inequality

$$\tilde{\mathbf{C}} \geq \mathbf{C} \text{ or } \tilde{\mathbf{C}} - \mathbf{C} \geq \mathbf{0}$$

states that $\tilde{\mathbf{C}} - \mathbf{C}$ is a positive semidefinite matrix. The identity matrix is denoted by $\mathbf{I}$.

II. PROBLEM STATEMENT AND THE FEDERATED KALMAN FILTER

We commence our discussion with the corresponding linear estimation problem. The state to be estimated is characterized by a discrete-time process model

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

with system matrix $\mathbf{A}_k$ and zero-mean white process noise $\underline{\boldsymbol{w}}_k$, which has the error covariance matrix $\mathbf{C}_k^{\boldsymbol{w}}$. The state is observed by a network of N sensor nodes. The sensor model of node i is defined by

$$\underline{\boldsymbol{z}}_k^i = \mathbf{H}_k^i \, \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k^i \ ,$$

where $\mathbf{H}_k^i$ is the measurement matrix and $\underline{\boldsymbol{v}}_k^i$ denotes the zero-mean white sensor noise with error covariance matrix $\mathbf{C}_k^{\boldsymbol{v},i}$. In a centralized Kalman filter, the data sink collects all measurements at each time step k, updates the estimate in the filtering step, and computes a predicted estimate for the next filtering step. Hence, data needs to be transmitted each time step. In order to reduce the communication rate, each node should compute an estimate of the state based on its own sensor observations. The estimate comprises the entire local measurement history, such that a less frequent or even an on-demand data transmission is sufficient.

However, the first problem to solve is the initialization of the local estimators. Let $(\underline{\hat{x}}_0^{\mathrm{p}}, \mathbf{C}_0^{\mathrm{p}})$ represent the prior knowledge about the state. Each local estimator could be initialized according to

$$\begin{bmatrix} \underline{\hat{x}}_0^{\mathrm{p},1} \\ \vdots \\ \underline{\hat{x}}_0^{\mathrm{p},N} \end{bmatrix} := \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} \underline{\hat{x}}_0^{\mathrm{p}} \ , \tag{1}$$

which results into the fully occupied MSE matrix

$$\mathrm{E}\left\{ \left(\begin{bmatrix} \underline{\hat{x}}_0^{\mathrm{p},1} \\ \vdots \\ \underline{\hat{x}}_0^{\mathrm{p},N} \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} \underline{\boldsymbol{x}}_0 \right) \begin{pmatrix} \cdots \end{pmatrix}^{\mathrm{T}} \right\} = \begin{bmatrix} \mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{C}_0^{\mathrm{p}} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{C}_0^{\mathrm{p}} \end{bmatrix}$$

expressing that each local estimator operates on a copy of the available prior information. The strong cross-correlations give rise to two problems: In the first place, cross-correlations need to be stored and updated somewhere. In the second place, the nodes cannot operate independently since, due to cross-correlations, a measurement update at sensor i affects each other local estimate. The federated Kalman filter (FKF) [12]–[14] solves this issue by replacing the MSE matrix with the upper bound

$$\begin{bmatrix} \frac{1}{\omega_1}\mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{1}{\omega_N}\mathbf{C}_0^{\mathrm{p}} \end{bmatrix} \geq \begin{bmatrix} \mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{C}_0^{\mathrm{p}} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{C}_0^{\mathrm{p}} \end{bmatrix} \tag{2}$$

with $\omega_1 + \ldots + \omega_N = 1$ and $\omega_i > 0$. This inflation method guarantees covariance-consistent fusion results and is also used in the covariance bounds algorithm [24], [25], which is an alternative formulation of the CI algorithm. However, in this particular case, fusion of the corresponding local estimates $(\underline{\hat{x}}_0^{\mathrm{p},i}, \mathbf{C}_0^{\mathrm{p},i}) := (\underline{\hat{x}}_0^{\mathrm{p}}, \frac{1}{\omega_i}\mathbf{C}_0^{\mathrm{p}})$ yields the original prior $(\underline{\hat{x}}_0^{\mathrm{p}}, \mathbf{C}_0^{\mathrm{p}})$, i.e.,

$$\begin{aligned} \underline{\hat{x}}_0^{\mathrm{fus}} &= \mathbf{C}_0^{\mathrm{fus}}\big((\mathbf{C}_0^{\mathrm{p},1})^{-1}\underline{\hat{x}}_0^{\mathrm{p},1} + \ldots + (\mathbf{C}_0^{\mathrm{p},N})^{-1}\underline{\hat{x}}_0^{\mathrm{p},N}\big) \\ &= \mathbf{C}_0^{\mathrm{fus}}\big(\omega_1(\mathbf{C}_0^{\mathrm{p}})^{-1}\underline{\hat{x}}_0^{\mathrm{p}} + \ldots \omega_N(\mathbf{C}_0^{\mathrm{p}})^{-1}\underline{\hat{x}}_0^{\mathrm{p}}\big) = \underline{\hat{x}}_0^{\mathrm{p}} \ , \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{C}_0^{\mathrm{fus}} &= \big((\mathbf{C}_0^{\mathrm{p},1})^{-1} + \ldots + (\mathbf{C}_0^{\mathrm{p},N})^{-1}\big)^{-1} \\ &= \big((\omega_1 + \ldots + \omega_N)(\mathbf{C}_0^{\mathrm{p}})^{-1}\big)^{-1} = \mathbf{C}_0^{\mathrm{p}} \ . \end{aligned} \tag{4}$$

Due to the (block) diagonal structure of (2), the Kalman filtering step for a measurement $\underline{\hat{z}}_k^i$ can be carried out locally. More precisely, the local prior or predicted estimate $(\underline{\hat{x}}_k^{\mathrm{p},i}, \mathbf{C}_k^{\mathrm{p},i})$ can

be updated to $(\hat{\underline{x}}_k^{\mathrm{e},i}, \mathbf{C}_k^{\mathrm{e},i})$ without affecting any other local estimate. Furthermore, the filtering step preserves the diagonal structure of the joint error covariance matrix. This stands in stark contrast to the prediction step, where the system model

$$\begin{bmatrix} \underline{\boldsymbol{x}}_{k+1}^{\mathrm{p},1} \\ \vdots \\ \underline{\boldsymbol{x}}_{k+1}^{\mathrm{p},N} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_k & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{A}_k \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{x}}_k^{\mathrm{p},1} \\ \vdots \\ \underline{\boldsymbol{x}}_k^{\mathrm{p},N} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} \underline{\boldsymbol{w}}_k$$

is applied to each "copy" of the state $\underline{\boldsymbol{x}}_k$, but where each row is affected by the same process noise $\underline{\boldsymbol{w}}_k$. In order to keep the joint cross-covariance matrix diagonal, the joint process noise error covariance matrix must be diagonal as well, which is achieved by the inflated matrix

$$\begin{bmatrix} \frac{1}{\omega_1}\mathbf{C}_k^{\boldsymbol{w}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{1}{\omega_N}\mathbf{C}_k^{\boldsymbol{w}} \end{bmatrix} \geq \begin{bmatrix} \mathbf{C}_k^{\boldsymbol{w}} & \cdots & \mathbf{C}_k^{\boldsymbol{w}} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_k^{\boldsymbol{w}} & \cdots & \mathbf{C}_k^{\boldsymbol{w}} \end{bmatrix} . \quad (5)$$

Each node i can then compute its local predicted estimate by means of

$$\hat{\underline{x}}_{k+1}^{\mathrm{p},i} = \mathbf{A}_k \, \hat{\underline{x}}_k^{\mathrm{e},i}$$

and

$$\mathbf{C}_{k+1}^{\mathrm{p},i} = \mathbf{A}_k \mathbf{C}_k^{\mathrm{e},i} \mathbf{A}_k^{\mathrm{T}} + \tfrac{1}{\omega_1}\mathbf{C}_k^{\boldsymbol{w}} \ .$$

A common choice for the inflation parameters is $\omega_i = \frac{1}{N}$, where N is the number of nodes in the network. In this case, the FKF is even in the position to provide optimal linear MMSE estimates if the local estimates are fused by (3) and (4) after each filtering step and are reinitialized with $(\hat{\underline{x}}_k^{\mathrm{fus}}, N\mathbf{C}_k^{\mathrm{fus}})$. If the estimates are fused after multiple local filtering and prediction steps, the FKF provides at least conservatively suboptimal fusion results. In summary, the local estimators place less trust in the propagated state estimates and rely more on their sensor readings. In doing so, the fusion center, i.e., the data sink, is prevented from being overconfident. The remainder of this paper unveils a generalization of the FKF to nonlinear estimation problems.

III. Nonlinear Distributed Estimation

Nonlinear state estimation is in itself an arduous task, but in networked systems it becomes even more challenging. An estimate of the uncertain state $\underline{\boldsymbol{x}}_k$ at time step k is represented by the conditional probability density function

$$f^{\mathrm{e}}(\underline{x}_k) := f(\underline{x}_k | \mathcal{Z}_{0:k}) = f(\underline{x}_k | \mathcal{Z}_0, \ldots, \mathcal{Z}_k) \ ,$$

where the set $\mathcal{Z}_l$ contains all sensor observations at time step l. The process model is given by any nonlinear function

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{w}}_k) \ , \quad (6)$$

where the process noise $\underline{\boldsymbol{w}}_k$ is characterized by a probability density $f_{\boldsymbol{w}}$. The sensor observations of node i are related to the system state by the model

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

where $\underline{\boldsymbol{v}}_k^i$ has the probability density $f_{\boldsymbol{v},i}$. The errors $\{\underline{\boldsymbol{w}}_k\}_{k\in\mathbb{N}}$ and $\{\underline{\boldsymbol{v}}_k^i\}_{k\in\mathbb{N}}$ are mutually independent and white. For the prediction step, the Chapman-Kolmogorov integral

$$f^{\mathrm{p}}(\underline{x}_{k+1}) := f(\underline{x}_{k+1} | \mathcal{Z}_{0:k}) = \int_{\mathbb{R}^n} f(\underline{x}_{k+1} | \underline{x}_k) f^{\mathrm{e}}(\underline{x}_k) \, \mathrm{d}\underline{x}_k \quad (7)$$

has to be computed. In its most general definition, the transition density can be written as the integral

$$f(\underline{x}_{k+1} | \underline{x}_k) = \int_{\mathbb{R}^n} \delta\big(\underline{x}_{k+1} - \underline{a}_k(\underline{x}_k, \underline{w}_k)\big) f_{\boldsymbol{w}}(\underline{w}_k) \, \mathrm{d}\underline{w}_k \ . \quad (8)$$

For a set of observations $\mathcal{Z}_k = \{\hat{\underline{z}}_k^1, \ldots, \hat{\underline{z}}_k^N\}$, the update step is performed by means of Bayes' rule for probability densities

$$\begin{aligned} f^{\mathrm{e}}(\underline{x}_k) &= C^{-1} \, f(\mathcal{Z}_k | \underline{x}_k) f^{\mathrm{p}}(\underline{x}_k) \\ &= C^{-1} \, f(\hat{\underline{z}}_k^1 | \underline{x}_k) \cdot \ldots \cdot f(\hat{\underline{z}}_k^N | \underline{x}_k) f^{\mathrm{p}}(\underline{x}_k) \ , \end{aligned} \quad (9)$$

where the last product of likelihoods is a consequence from the conditional independence of the measurements given the state.

For the purpose of implementing a Bayesian estimator within a sensor network, the joint state (1) has to be considered, and a special advantageous representation of the joint density $f^{\mathrm{e}}\big(\underline{x}_k^1, \ldots, \underline{x}_k^N\big)$ is aspired. The update step (9) has already indicated that a product

$$f^{\mathrm{e}}\big(\underline{x}_k^1, \ldots, \underline{x}_k^N\big) = f^{\mathrm{e},1}(\underline{x}_k^1) \cdot \ldots \cdot f^{\mathrm{e},N}(\underline{x}_k^N) \quad (10)$$

of independent densities appears to be particularly suitable since every local estimate $f^{\mathrm{e},i}(\underline{x}_k^i)$ can then be processed with the individual likelihood $f(\hat{\underline{z}}_k^i | \underline{x}_k)$ without affecting other local estimates. Again, the prediction step does not preserve the product form (10). In the joint state space $[\underline{\boldsymbol{x}}_k^1, \ldots, \underline{\boldsymbol{x}}_k^N]^{\mathrm{T}} = [\underline{\boldsymbol{x}}_k, \ldots, \underline{\boldsymbol{x}}_k]^{\mathrm{T}}$, the process model has the form

$$\begin{bmatrix} \underline{\boldsymbol{x}}_{k+1}^1 \\ \vdots \\ \underline{\boldsymbol{x}}_{k+1}^N \end{bmatrix} = \tilde{\underline{a}}_k \left(\begin{bmatrix} \underline{\boldsymbol{x}}_k^1 \\ \vdots \\ \underline{\boldsymbol{x}}_k^N \end{bmatrix}, \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} \underline{\boldsymbol{w}}_k \right) := \begin{bmatrix} \underline{a}_k(\underline{\boldsymbol{x}}_k^1, \underline{\boldsymbol{w}}_k) \\ \vdots \\ \underline{a}_k(\underline{\boldsymbol{x}}_k^N, \underline{\boldsymbol{w}}_k) \end{bmatrix} ,$$

and the predicted joint density then yields

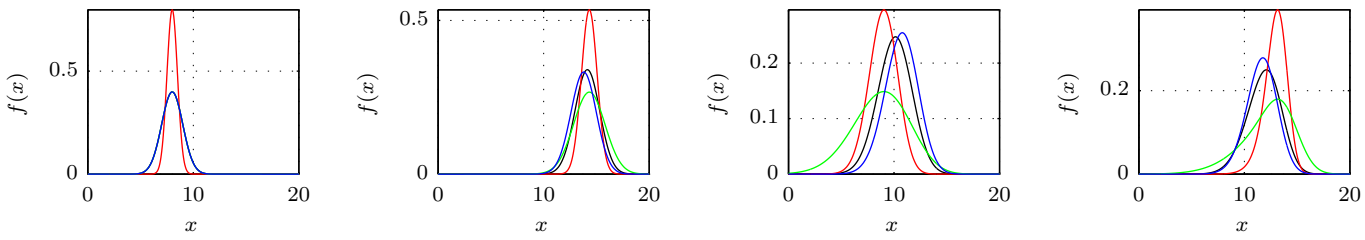
$$\begin{aligned} &f^{\mathrm{p}}(\underline{x}_{k+1}^1, \ldots, \underline{x}_{k+1}^N) = \\ &\int_{\mathbb{R}^n} \cdots \int_{\mathbb{R}^n} f(\underline{x}_{k+1}^1, \ldots, \underline{x}_{k+1}^N \,|\, \underline{x}_k^1, \ldots, \underline{x}_k^N) \cdot \\ &\qquad\qquad f^{\mathrm{e}}(\underline{x}_k^1, \ldots, \underline{x}_k^N) \, \mathrm{d}\underline{x}_k^1 \ldots \mathrm{d}\underline{x}_k^N \ . \end{aligned} \quad (11)$$

Even if $f^{\mathrm{e}}(\underline{x}_k^1, \ldots, \underline{x}_k^N)$ is given in the form (10), the common process noise prevents a product representation of the prediction result (11). In the following sections, we hence derive a conservative prediction technique.

IV. Nonlinear Federated Filtering for Additive Gaussian Noise

In order to apply the FKF in nonlinear estimation problems, linearized system and measurement models can be considered [26], or other linearization techniques [27] can be exploited. These approaches share the common idea to represent estimates by their first two moments. However, mean and estimate variance often do not suffice to characterize the uncertain state, and they can even be deceptive especially when the actual probability density of the state estimate is multi-modal. Hence, we aspire to derive a generic solution that is applicable to arbitrary density representations.

As outlined in the preceding section, a product representation of the joint density appears to be particularly promising for nonlinear federated filtering. Instead of a linear MMSE estimator, the standard Kalman filter can also be regarded as a closed-form solution to the Bayesian estimation problem (7)



(a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4.

Figure 1. The black density (—) denotes the estimation result of a centralized estimator—the optimal result. The red density (—) is the result of a naive fusion of the four local estimates at each time step. The EMD-fusion result of the local estimates is drawn green (—) and the result of the nonlinear federated filtering algorithm is drawn in blue (—).

and (9) when models are linear and random variables are Gaussian. In this line, the prior knowledge $(\hat{\underline{x}}_0^{\mathrm{p}}, \mathbf{C}_0^{\mathrm{p}})$ is equivalent to

$$f^{\mathrm{p}}(\underline{x}_0) = \mathcal{N}(\underline{x}_0; \hat{\underline{x}}_0^{\mathrm{p}}, \mathbf{C}_0^{\mathrm{p}}) \ .$$

The FKF in Sec. II initializes the joint state estimate with $[\hat{\underline{x}}_0^{\mathrm{p}}, \dots, \hat{\underline{x}}_0^{\mathrm{p}}]^{\mathrm{T}}$ and the inflated covariance matrix (2). Apparently, the corresponding density $f^{\mathrm{p}}(\underline{x}_0^1, \dots, \underline{x}_0^N)$ has already the product form

$$\begin{aligned}
&\mathcal{N}\left(\begin{bmatrix}\underline{x}_0^1\\ \vdots \\ \underline{x}_0^N\end{bmatrix}; \begin{bmatrix}\hat{\underline{x}}_0\\ \vdots \\ \hat{\underline{x}}_0\end{bmatrix}, \begin{bmatrix}\frac{1}{\omega_1}\mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{1}{\omega_N}\mathbf{C}_0^{\mathrm{p}}\end{bmatrix}\right)\\
&= \mathcal{N}\left(\underline{x}_0^1; \hat{\underline{x}}_0, \tfrac{1}{\omega_1}\mathbf{C}_0^1\right) \cdot \ldots \cdot \mathcal{N}\left(\underline{x}_0^N; \hat{\underline{x}}_0, \tfrac{1}{\omega_N}\mathbf{C}_0^N\right)\\
&= c_1 \left(\mathcal{N}\left(\underline{x}_0^1; \hat{\underline{x}}_0, \mathbf{C}_0^1\right)\right)^{\omega_1} \cdot \ldots \cdot c_N \left(\mathcal{N}\left(\underline{x}_0^N; \hat{\underline{x}}_0, \mathbf{C}_0^N\right)\right)^{\omega_N}
\end{aligned}$$

with normalization constants $c_1, \dots, c_N$. This means that the prior density is written as

$$f^{\mathrm{p}}(\underline{x}_0) = \left(f^{\mathrm{p}}(\underline{x}_0)\right)^{\omega_1} \cdot \ldots \cdot \left(f^{\mathrm{p}}(\underline{x}_0)\right)^{\omega_N}$$

and the local estimates are thus initialized by

$$f^{\mathrm{p},i}(\underline{x}_0) := \frac{\left(f^{\mathrm{p}}(\underline{x}_0)\right)^{\omega_i}}{\int_{\mathbb{R}^{n_x}} \left(f^{\mathrm{p}}(\underline{x}_0)\right)^{\omega_i} \, \mathrm{d}\underline{x}_0} \ . \tag{12}$$

Evidently, this technique can directly be generalized to arbitrary non-Gaussian densities.

As mentioned before, local filtering steps do not harm the product representation, and, hence, we focus on the prediction step. For the following considerations, we first confine ourselves to models

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k) + \underline{\boldsymbol{w}}_k \tag{13}$$

with additive Gaussian noise $f_{\boldsymbol{w}} = \mathcal{N}(\underline{w}_k; \underline{0}, \mathbf{C}_k^{\boldsymbol{w}})$. In this particular case, the transition density can be expressed in terms of the density $f_{\boldsymbol{w}}$ according to

$$\begin{aligned}
f(\underline{x}_{k+1} \,|\, \underline{x}_k) &= f_{\boldsymbol{w}}(\underline{x}_{k+1} - \underline{a}_k(\underline{x}_k))\\
&= \mathcal{N}\left(\underline{x}_{k+1} - \underline{a}_k(\underline{x}_k); \underline{0}, \mathbf{C}_k^{\boldsymbol{w}}\right) \ .
\end{aligned} \tag{14}$$

Analogously, the process model for the joint state space yields

$$\begin{bmatrix}\underline{\boldsymbol{x}}_{k+1}^1\\ \vdots \\ \underline{\boldsymbol{x}}_{k+1}^N\end{bmatrix} = \begin{bmatrix}\underline{a}_k(\underline{\boldsymbol{x}}_k^1)\\ \vdots \\ \underline{a}_k(\underline{\boldsymbol{x}}_k^N)\end{bmatrix} + \begin{bmatrix}\mathbf{I}\\ \vdots \\ \mathbf{I}\end{bmatrix} \underline{\boldsymbol{w}}_k \ ,$$

which means that the model (13) has to be applied to each "copy" of the state and the process noise terms are fully correlated. As a result, the transition density (14) has to be generalized to the joint transition density

$$\begin{aligned}
&f(\underline{x}_{k+1}^1, \dots, \underline{x}_{k+1}^N \,|\, \underline{x}_k^1, \dots, \underline{x}_k^N) =\\
&\mathcal{N}\left(\begin{bmatrix}\underline{x}_{k+1}^1\\ \vdots \\ \underline{x}_{k+1}^N\end{bmatrix} - \begin{bmatrix}\underline{a}_k(\underline{x}_k^1)\\ \vdots \\ \underline{a}_k(\underline{x}_k^N)\end{bmatrix}; \begin{bmatrix}\underline{0}\\ \vdots \\ \underline{0}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_k^{\boldsymbol{w}} & \cdots & \mathbf{C}_k^{\boldsymbol{w}}\\ \vdots & \ddots & \vdots \\ \mathbf{C}_k^{\boldsymbol{w}} & \cdots & \mathbf{C}_k^{\boldsymbol{w}}\end{bmatrix}\right) .
\end{aligned}$$

The linear federated Kalman filter employs the inflated process noise matrix (5) in order to decouple the local prediction steps. In the same fashion, we can decompose the joint transition density into the product

$$\begin{aligned}
&\tilde{f}(\underline{x}_{k+1}^1, \dots, \underline{x}_{k+1}^N \,|\, \underline{x}_k^1, \dots, \underline{x}_k^N) =\\
&\mathcal{N}\left(\begin{bmatrix}\underline{x}_{k+1}^1\\ \vdots \\ \underline{x}_{k+1}^N\end{bmatrix} - \begin{bmatrix}\underline{a}_k(\underline{x}_k^1)\\ \vdots \\ \underline{a}_k(\underline{x}_k^N)\end{bmatrix}; \begin{bmatrix}\underline{0}\\ \vdots \\ \underline{0}\end{bmatrix}, \begin{bmatrix}\frac{1}{\omega_1}\mathbf{C}_k^{\boldsymbol{w}} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{1}{\omega_N}\mathbf{C}_k^{\boldsymbol{w}}\end{bmatrix}\right)\\
&= \prod_{i=1}^{N} \underbrace{\mathcal{N}\left(\underline{x}_{k+1}^i - \underline{a}_k(\underline{x}_k^i); \underline{0}, \tfrac{1}{\omega_i}\mathbf{C}_k^{\boldsymbol{w}}\right)}_{=: \tilde{f}(\underline{x}_{k+1}^i \,|\, \underline{x}_k^i)}
\end{aligned}$$

by replacing the process noise covariance matrix with the bound (5). Given that the current estimate f^{e} has the product form (10), the prediction step (11), i.e.,

$$\begin{aligned}
&\tilde{f}^{\mathrm{p}}(\underline{x}_{k+1}^1, \dots, \underline{x}_{k+1}^N) =\\
&\int_{\mathbb{R}^{n_x}} \cdots \int_{\mathbb{R}^{n_x}} \tilde{f}(\underline{x}_{k+1}^1 \,|\, \underline{x}_k^1) \dots \tilde{f}(\underline{x}_{k+1}^N \,|\, \underline{x}_k^N) \cdot\\
&\qquad\qquad f^{\mathrm{e},1}(\underline{x}_k^1) \dots f^{\mathrm{e},N}(\underline{x}_k^N) \, \mathrm{d}\underline{x}_k^1 \dots \mathrm{d}\underline{x}_k^N\\
&= \prod_{i=1}^{N} \underbrace{\int_{\mathbb{R}^{n_x}} \tilde{f}(\underline{x}_{k+1}^i \,|\, \underline{x}_k^i) f^{\mathrm{e},i}(\underline{x}_k^i) \, \mathrm{d}\underline{x}_k^i}_{= f^{\mathrm{p},i}(\underline{x}_{k+1}^i)} \ ,
\end{aligned}$$

now preserves the product form.

At each time step k, the local estimates $f^{\mathrm{e},i}$ can simply be fused by computing the product

$$f^{\mathrm{fus}}(\underline{x}_k) = C^{-1} f^{\mathrm{e},1}(\underline{x}_k) \cdot \ldots \cdot f^{\mathrm{e},N}(\underline{x}_k) \ ,$$

where C is a normalization constant. More precisely, this means that the local estimates are modeled to be independent, such as it is done by the FKF. The following example illustrates the proposed nonlinear federated filter.

Example: Four Sensor Nodes
A one-dimensional state is tracked by four sensors over four time steps. The initial state is $x_0 = 8$ and the process model

$$\boldsymbol{x}_{k+1} = a(\boldsymbol{x}_k) := 0.23(16\boldsymbol{x} - \boldsymbol{x}^2) + \boldsymbol{w}_k$$

is nonlinear, where the additive zero-mean Gaussian noise $\boldsymbol{w}_k$ has the variance $C_k^{\boldsymbol{w}} = 0.16$. The state is measured directly by each sensor, i.e., $h^i(x_k) = x_k$. The measurements are affected by an additive zero-mean noise with variance $C_k^{\boldsymbol{v},i} = 16$, $i = 1, \ldots, 4$. In Fig. 1, the fusion results are depicted. An optimal strategy is a central Bayesian estimator that collects all measurements at each time step (black density). The red density is obtained when the sensor nodes run local estimators and fuse their results naively in a data sink by multiplying the densities pointwise. Apparently, the results are rather overconfident. The blue density is a result of the nonlinear federated filtering scheme, where each node inflates its local process noise by factor 4 ($\omega_i = 0.25$). The fusion can again be done naively, but the result at each step is close to the optimal one. The green density corresponds to the EMD fusion rule [18] applied to the local estimates (without inflation of the process noise). It is a very conservative result.

V. General Solution: Nonlinear Federated Filtering for Arbitrary Models

By following the same train of thought that has led to the EMD fusion rule, the nonlinear federated filter can now be generalized to arbitrary system models (6) with non-Gaussian process noise. For additive Gaussian process noise as in (13), the transition density derived in the last section can be rewritten to

$$\begin{aligned}
\tilde{f}(\underline{x}_{k+1}^i \,|\, \underline{x}_k^i) &= \mathcal{N}\Big(\underline{x}_{k+1}^i - \underline{a}_k(\underline{x}_k^i); \underline{0}, \tfrac{1}{\omega_i}\mathbf{C}_k^{\boldsymbol{w}}\Big) \\
&= c^i \Big(\mathcal{N}\Big(\underline{x}_{k+1}^i - \underline{a}_k(\underline{x}_k^i); \underline{0}, \mathbf{C}_k^{\boldsymbol{w}}\Big)\Big)^{\omega_i} \\
&= c^i \Big(f_{\boldsymbol{w}}(\underline{x}_{k+1}^i - \underline{a}_k(\underline{x}_k^i))\Big)^{\omega_i} \\
&\overset{(14)}{=} c^i f^{\omega_i}(\underline{x}_{k+1}^i \,|\, \underline{x}_k^i) \ ,
\end{aligned}$$

where c^i is a normalization factor. Accordingly, an arbitrary joint transition density $f(\underline{x}_{k+1}^1, \ldots, \underline{x}_{k+1}^N \,|\, \underline{x}_k^1, \ldots, \underline{x}_k^N)$ can be conservatively represented by an exponential mixture density

$$\begin{aligned}
&f_{\mathrm{EMD}}(\underline{x}_{k+1}^1, \ldots, \underline{x}_{k+1}^N \,|\, \underline{x}_k^1, \ldots, \underline{x}_k^N) \\
&\quad = c^1 f^{\omega_1}(\underline{x}_{k+1}^1 \,|\, \underline{x}_k^1) \cdot \ldots \cdot c^N f^{\omega_N}(\underline{x}_{k+1}^N \,|\, \underline{x}_k^N)
\end{aligned} \tag{15}$$

with $\sum_{i=1}^N \omega_i = 1$, $\omega_i \geq 0$, and c^i denoting the corresponding normalization constants. In contrast to the EMD fusion rule [18], the EMD representation is here used to decompose a single density into several densities and not for fusing several densities into one density. Each factor $f^{\omega_i}(\underline{x}_{k+1}^i \,|\, \underline{x}_k^i)$ can locally be computed for any system model (6) so as to perform the local prediction step

$$f^{\mathrm{p}}(\underline{x}_{k+1}^i) = \int_{\mathbb{R}^{n_x}} c^i f^{\omega_i}(\underline{x}_{k+1}^i \,|\, \underline{x}_k^i) \cdot f^{\mathrm{e},i}(\underline{x}_k^i) \ \mathrm{d}\underline{x}_k^i$$

at sensor node i. According to the definition in [17], each component of the conditional EMD density (15) is a conservative approximation of the local transition density (8).

In summary, each node can now operate independently and in parallel. The local estimators can be initialized by (12) and, for each prediction step, the joint transition density in (11) is conservatively substituted by the product (15) of independent transition densities.

VI. Simulations

In order to discuss the presented idea, we consider the problem of tracking a mobile object. The state is characterized by the object's 2D position and orientation with $\underline{x} = [x^1, x^2, \phi]^{\mathrm{T}}$. The discrete-time motion model

$$\underline{x}_{k+1} = \begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k^1 \\ x_k^2 \\ \phi_k \end{bmatrix} + T\,\mathbf{R}(\phi_k) \begin{bmatrix} u_k \\ 0 \\ \frac{u_k}{l}\tan(\alpha_k) \end{bmatrix}$$

corresponds to the kinematics of an idealized bicycle, where $l = 0.5$ is the distance between the wheels, the rotation matrix $\mathbf{R}(\phi_k)$ characterizes the orientation in the x^1-x^2-plane, and u_k and α_k are control inputs for velocity and orientation, respectively. The sampling time is set to $T = 0.1$. The velocity $u_k = 27$ is constant, but the steering angle $\alpha_k = 0$ is perturbed by a zero-mean Gaussian noise with variance $C_k^{\alpha} = 3$. Hence, the process noise nonlinearly affects the system model. The position is estimated by means of four closely positioned distance sensors, that are located at $P^1 = [10, 5]^{\mathrm{T}}$, $P^2 = [12, 10]^{\mathrm{T}}$, $P^3 = [14, 6.1]^{\mathrm{T}}$, and $P^4 = [8, 7]^{\mathrm{T}}$. The sensor models are

$$\boldsymbol{z}_k^i = \sqrt{(\boldsymbol{x}_k^1 - P_{x^1}^i)^2 + (\boldsymbol{x}_k^2 - P_{x^2}^i)^2} + \boldsymbol{v}_k^i \ ,$$

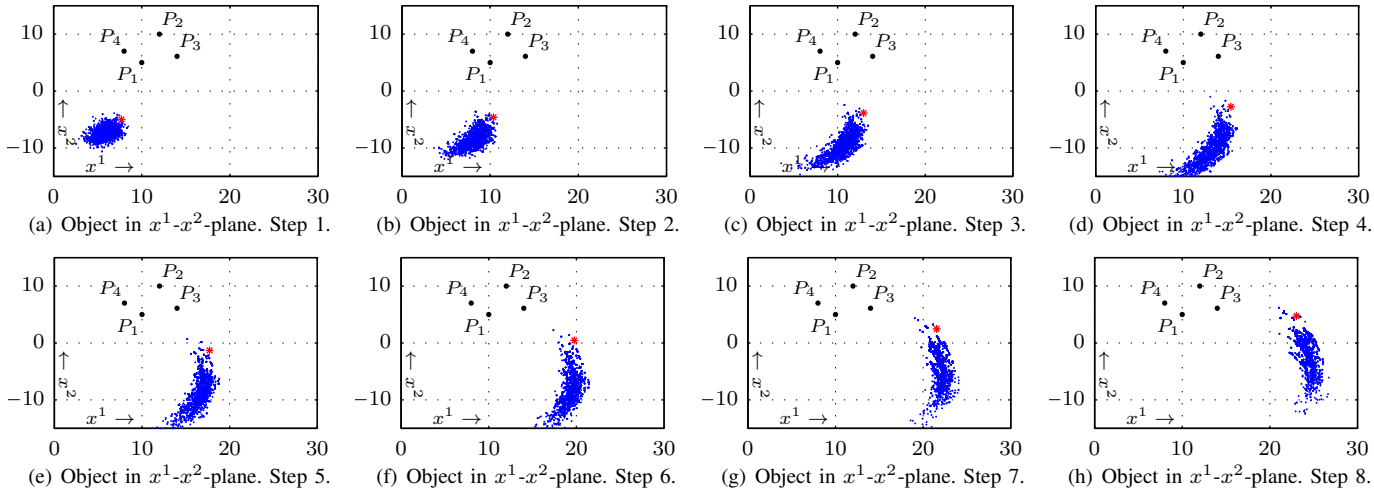
where $\boldsymbol{v}_k^i$ is a high observation noise with variance $C_k^{\boldsymbol{v},i} = 64$. The state is estimated by means of a simple particle filter [28].

The trajectory has been plotted over eight time steps in Figures 2 - 4. A centralized solution has been employed for computing the estimation results in Fig. 2. Only, the data sink runs an estimator and collects all measurements at each time step and, hence, is able to provide the optimal Bayesian estimation result. A naive solution is depicted in Fig. 3, where particle filters are implemented locally on each node. The local results are fused at each time step at the data sink, where possible dependencies are simply ignored. For the purpose of fusing particle representations, the kernel $\mathcal{N}(\underline{x}_k; \underline{0}, \mathrm{diag}([0.1, 0.1, 0.005])$ has been employed to obtain continuous representations and to compute the product of the local estimates. Evidently, the track is lost and the fusion result is by far too confident. By contrast, Fig. 4 shows close-to-optimal fusion results, which can be achieved when the local process noise is increased by factor 4 ($\omega_i = 0.25$)—the number of nodes. More precisely, the EMD transition density (15) has been employed, in order to keep the local tracks independent of each other.

VII. Conclusions and Outlook

Distributed state estimation and tracking applications generally encounter the problem that data have to be fused that have been computed independently but are dependent. For distributed linear estimation problems, a bunch of solutions is available. The most notable and flexible algorithm is covariance intersection. But, in order to reduce conservatism, other less conservative methods might be preferable. If data fusion takes place only at the data sink, common process noise can be pinpointed as the only cause of dependencies. The federated Kalman filter is then in the position to directly counteract the common process noise by increasing the locally modeled noise proportionally to the number of participating nodes. As a consequence, less conservative fusion results are ensured.

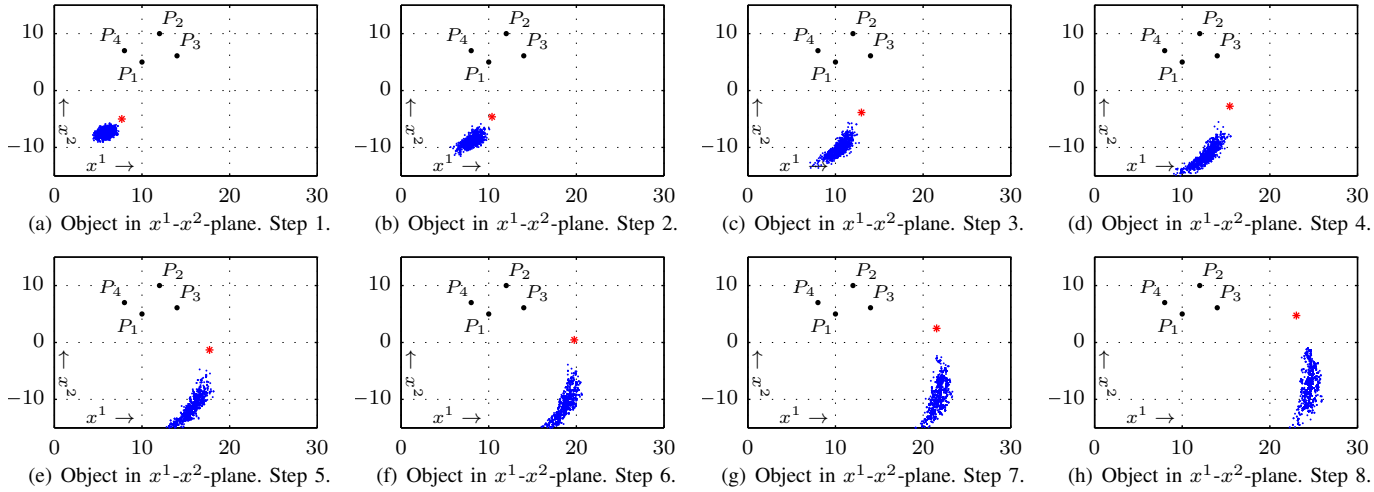
The exponential mixture fusion rule embodies a generalization of covariance intersection to nonlinear estimation



(a) Object in x^1-x^2-plane. Step 1. (b) Object in x^1-x^2-plane. Step 2. (c) Object in x^1-x^2-plane. Step 3. (d) Object in x^1-x^2-plane. Step 4.

(e) Object in x^1-x^2-plane. Step 5. (f) Object in x^1-x^2-plane. Step 6. (g) Object in x^1-x^2-plane. Step 7. (h) Object in x^1-x^2-plane. Step 8.

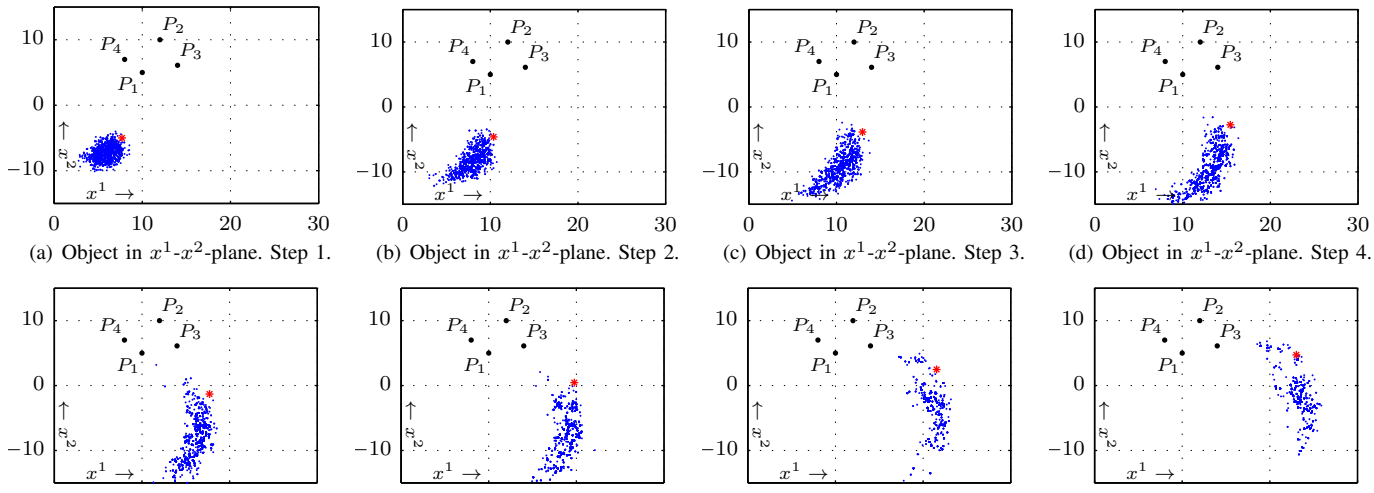
Figure 2. Particle filter result. Centralized optimal solution. Red star represents the object's position.



(a) Object in x^1-x^2-plane. Step 1. (b) Object in x^1-x^2-plane. Step 2. (c) Object in x^1-x^2-plane. Step 3. (d) Object in x^1-x^2-plane. Step 4.

(e) Object in x^1-x^2-plane. Step 5. (f) Object in x^1-x^2-plane. Step 6. (g) Object in x^1-x^2-plane. Step 7. (h) Object in x^1-x^2-plane. Step 8.

Figure 3. Particle filter result after fusion of local estimates. Dependencies have been ignored. Red star represents the object's position.



(a) Object in x^1-x^2-plane. Step 1. (b) Object in x^1-x^2-plane. Step 2. (c) Object in x^1-x^2-plane. Step 3. (d) Object in x^1-x^2-plane. Step 4.

(e) Object in x^1-x^2-plane. Step 5. (f) Object in x^1-x^2-plane. Step 6. (g) Object in x^1-x^2-plane. Step 7. (h) Object in x^1-x^2-plane. Step 8.

Figure 4. Particle filter result after fusion of local estimates. Nonlinear federated filter has been employed. Red star represents the object's position.

problems and likewise is often too conservative. Therefore, a nonlinear counterpart to the federated Kalman filter has been derived in this paper that keeps the local tracks independent by conservatively substituting the joint transition density by an exponential mixture density. This technique also allows to circumvent dependencies induced by non-Gaussian common process noise. The nonlinear federated filter only bounds that part of the estimation problem that is responsible for dependencies and is hence less conservative than the exponential mixture fusion rule. Of course, in case that other sources of dependencies are present, the proposed approach entails the risk of being overconfident. However, its easy applicability within many state estimation systems is a very beneficial property. As demonstrated in the simulation, particle filter implementations can, for instance, easily be extended to distributed estimation problems.

A major direction of future work is optimal distributed Bayesian state estimation. In the linear case, the federated Kalman filter can be recognized as a predecessor of optimal distributed Kalman filters like [6], [7]. In the same way, the presented idea may provide the groundwork for optimal distributed nonlinear estimators.

ACKNOWLEDGMENT

This work was partially supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 “Self-organizing Sensor-Actuator-Networks”.

REFERENCES

[1] 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.

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

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

[4] Y. Bar-Shalom and L. Campo, “The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, no. 6, pp. 803–805, Nov. 1986.

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

[6] H. R. Hashemipour, S. Roy, and A. J. Laub, “Decentralized Structures for Parallel Kalman Filtering,” *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, Jan. 1988.

[7] W. Koch, “On Optimal Distributed Kalman Filtering and Retrodiction at Arbitrary Communication Rates for Maneuvering Targets,” in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, Aug. 2008.

[8] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME*, pp. 35–45, 1960.

[9] 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.

[10] S. J. Julier and J. K. Uhlmann, *Handbook of Multisensor Data Fusion: Theory and Practice*, 2nd ed. CRC Press, 2009, ch. General Decentralized Data Fusion with Covariance Intersection, pp. 319–343.

[11] 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.

[12] N. A. Carlson, “Federated Filter for Fault-tolerant Integrated Navigation Systems,” in *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS’88)*, Orlando, Florida, USA, 1988, pp. 110–119, record ‘Navigation into the 21st Century’ (IEEE Cat. No.88CH2675-7).

[13] N. A. Carlson, “Information-sharing Approach to Federated Kalman Filtering,” in *Proceedings of the IEEE Aerospace and Electronics Conference 1988 (NAECON 1988)*, Dayton, Ohio, USA, May 1988.

[14] N. A. Carlson, “Federated Square Root Filter for Decentralized Parallel Processors,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, pp. 517–525, 1990.

[15] R. P. S. Mahler, “Optimal/Robust Distributed Data Fusion: A Unified Approach,” in *Proceedings of SPIE*, vol. 4052, 2000.

[16] M. B. Hurley, “An Information Theoretic Justification for Covariance Intersection and its Generalization,” in *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*, Annapolis, Maryland, USA, Jul. 2002.

[17] T. Bailey, S. J. Julier, and G. Agamennoni, “On Conservative Fusion of Information with Unknown Non-Gaussian Dependence,” in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[18] S. J. Julier, T. Bailey, and J. K. Uhlmann, “Using Exponential Mixture Models for Suboptimal Distributed Data Fusion,” in *IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW 2006)*, Cambridge, United Kingdom, Sep. 2006, pp. 160–163.

[19] B. Noack, M. Baum, and U. D. Hanebeck, “Covariance Intersection in Nonlinear Estimation Based on Pseudo Gaussian Densities,” in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[20] M. Üney, D. E. Clark, and S. J. Julier, “Information Measures in Distributed Multitarget Tracking,” in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[21] D. Clark, S. J. Julier, R. P. S. Mahler, and B. Ristic, “Robust Multi-object Sensor Fusion with Unknown Correlations,” in *Sensor Signal Processing for Defence (SSPD 2010)*, London, United Kingdom, Sep. 2010.

[22] L.-L. Ong, M. Ridley, B. Upcroft, S. Kumar, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, “A Comparison of Probabilistic Representations for Decentralised Data Fusion,” in *Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference (ISSNIP 2005)*, Dec. 2005, pp. 187–192.

[23] 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.

[24] U. D. Hanebeck and K. Briechle, “New Results for Stochastic Prediction and Filtering with Unknown Correlations,” in *Proceedings of the 2001 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001)*, Baden–Baden, Germany, Aug. 2001, pp. 147–152.

[25] U. D. Hanebeck, K. Briechle, and J. Horn, “A Tight Bound for the Joint Covariance of Two Random Vectors with Unknown but Constrained Cross–Correlation,” in *Proceedings of the 2001 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001)*, Baden–Baden, Germany, Aug. 2001, pp. 85–90.

[26] A. Edelmayer and M. Miranda, “Federated Filtering for Fault-tolerant Estimation and Sensor Redundancy Management in Coupled Dynamics Distributed Systems,” in *Proceedings of the 2007 Mediterranean Conference on Control and Automation (MED 2007)*, Athens, Greece, Jun. 2007.

[27] B. Guan, X. Tang, and Q. Ge, “Federated Cubature Kalman Filter for Multi-sensor Information Fusion,” in *Proceedings of 2012 International Conference on Computer Technology and Science (ICCTS 2012)*, New Delhi, India, Aug. 2012.

[28] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.