On Nonlinear Track-to-track Fusion with Gaussian Mixtures

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
{benjamin.noack, marc.reinhardt, uwe.hanebeck}@ieee.org

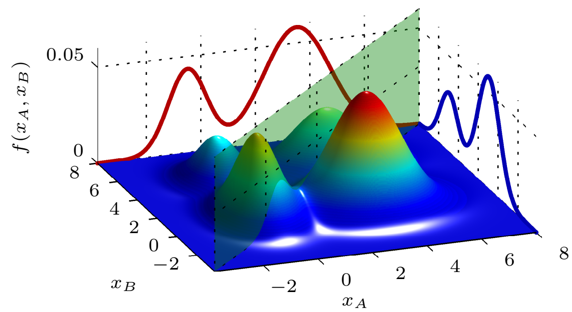
***Abstract*—The problem of fusing state estimates is encountered in many network-based multi-sensor applications. The majority of distributed state estimation algorithms are designed to provide multiple estimates on the same state, and track-to-track fusion then refers to the task of combining these estimates. While linear fusion only requires the joint cross-covariance matrix to be known, dependencies between estimates in nonlinear estimation problems have to be represented by high-dimensional probability density functions. In general, storing and keeping track of nonlinear dependencies is too cumbersome. However, this paper demonstrates that estimates represented by Gaussian mixtures prove to be an important exception to this rule. The dependency structure can as well be characterized in terms of a higher-dimensional Gaussian mixture. The different processing steps of distributed nonlinear state estimation, i.e., prediction, filtering, and fusion, are studied in light of the joint density representation. The presented concept is complemented with different simpler suboptimal representations of the dependency structure between Gaussian mixture densities.**

***Keywords*—Nonlinear estimation, distributed estimation, unknown dependencies, Gaussian mixtures.**

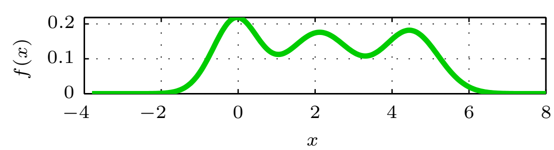
I. Introduction

Distributed implementations of state estimation algorithms [1], [2] become increasingly important in view of the trend towards network-based multi-sensor systems. Networks of multiple sensor devices offer several advantages over centralized systems including scalability, robustness to failures, and the ability to monitor large-scale phenomena. Instead of continuously transmitting measurements to a central estimation system, it is usually desired to compute track estimates of the state at the sensor nodes of a multi-sensor system or at local processing nodes in close proximity to the sensors and to transmit these estimates only when a fusion result is desired. The combination of locally computed estimates is referred to as track-to-track fusion [2]–[4].

The majority of studies on distributed estimation problems are concerned with linear process and sensor models [1], [3]–[6]. Linear state estimation problems generally prove to be more intuitive and handier than their nonlinear counterparts, and the Kalman filter algorithm [7] can be employed for the purpose of computing minimum mean-squared-error (MMSE) estimates. However, distributed or decentralized problem setups still render linear state estimation difficult. The local processing and subsequent fusion of Kalman filter estimates does not resemble a single centralized Kalman filter, which is fed with all measurement data at every time step, i.e., MMSE fusion does not correspond to the MMSE estimate given all available measurements. This discrepancy has been studied, for instance, in [4] and can be addressed by full-rate communication networks [8] or by employing global knowledge at the local processing nodes [5], [6].



(a) Joint density for two Gaussian mixtures that represent local estimates of the same state.



(b) Fusion result for the joint Gaussian mixture shown in Fig. (a). The green density corresponds to the density lying on the green plane.

Figure 1. Joint probability density function of the local red and blue estimates.

The major challenge in track-to-track fusion problems is the correct treatment of dependencies between the tracks to be fused. The general problem is that estimates though being computed independently cannot be deemed to be independent. The main reason for interdependencies resides in the fact that each locally computed estimate refers to the same state to be tracked, and hence, the same process noise is modeled multiple times [3]. A fusion algorithm must be in the position to account for possible dependencies in order to prevent erroneous or overconfident fusion results. Optimal solutions, such as [5], [9], often rely on stringent assumptions, which are often difficult to meet. An alternative direction is to employ suboptimal fusion strategies [10] that provide conservative fusion results irrespectively of the underlying dependencies.

In nonlinear distributed estimation problems, the treatment of dependencies between local estimates is even more complicated [11]–[14]. Here, dependencies are represented by joint probability density functions [15], [16]. This paper contributes a systematic derivation of the dependency structure between Gaussian mixture estimates, for which an example is shown in Fig. 1(a). Gaussian mixtures are a widely used concept in nonlinear estimation problems, e.g., for multi-target tracking [17] or in machine learning [18]. The major result of this paper is that the dependency structure itself is a Gaussian mixture, and it is studied how this Gaussian mixture is affected by the processing steps of the considered nonlinear estimator. Compared with linear estimation problems, it is demonstrated, which additional challenges have to be addressed in nonlinear distributed estimation problems. Also, strategies for suboptimal representations of the dependency structure are presented.

II. Preliminaries

Underlined variables $\underline{x} \in \mathbb{R}^{n_x}$ 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 $\mathbf{C}$. The pair $(\hat{\underline{x}}^{\mathrm{e}}, \mathbf{C}^{\mathrm{e}})$ represents an estimate $\hat{\underline{x}}$ of an uncertain state $\underline{\boldsymbol{x}}$ with error covariance matrix

$$\mathbf{C} = \mathrm{E}\left\{(\hat{\underline{x}} - \underline{\boldsymbol{x}})(\hat{\underline{x}} - \underline{\boldsymbol{x}})^{\mathrm{T}}\right\} \ .$$

A random quantity $\underline{\boldsymbol{x}}$ with $\underline{\boldsymbol{x}} \sim \mathcal{N}(\hat{\underline{x}}, \mathbf{C})$ is normally distributed, and the function $\mathcal{N}(\cdot; \hat{\underline{x}}, \mathbf{C})$ is the Gaussian density

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

with mean $\hat{\underline{x}}$ and covariance matrix $\mathbf{C}$. The scalar C is the normalization factor. The identity matrix is denoted by $\mathbf{I}$. The index $k \in \mathbb{N}$ is the time step; it is omitted if a fixed time step is considered. The superscript e is used for estimates and filtering results, and the superscript p is used for prior estimates and the results of the prediction step.

III. Linear Track-to-track Fusion

In this and the subsequent sections, we confine ourselves to the problem of fusing two track estimates stemming from sensor nodes A and B. This section summarizes the problem of linearly fusing Kalman filter estimates. For this purpose, the dependency structure after prediction, global and local filtering is analyzed.

A major benefit of using linear Kalman filter methods is that dependencies between two estimates $(\hat{\underline{x}}^{\mathrm{e}}_A, \mathbf{C}^{\mathrm{e}}_A)$ and $(\hat{\underline{x}}^{\mathrm{e}}_B, \mathbf{C}^{\mathrm{e}}_B)$ are given by the cross-covariance matrix

$$\mathbf{C}^{\mathrm{e}}_{AB} = \mathrm{E}\left\{(\hat{\underline{x}}^{\mathrm{e}}_A - \underline{\boldsymbol{x}})(\hat{\underline{x}}^{\mathrm{e}}_B - \underline{\boldsymbol{x}})^{\mathrm{T}}\right\}$$

of the estimation errors. The two estimates can be concatenated together into the joint state estimate $\hat{\underline{x}}^{\mathrm{e}}_{\mathrm{J}} = [(\hat{\underline{x}}^{\mathrm{e}}_A)^{\mathrm{T}}, (\hat{\underline{x}}^{\mathrm{e}}_A)^{\mathrm{T}}]^{\mathrm{T}}$, which has the joint error covariance matrix

$$\begin{aligned}\mathbf{C}^{\mathrm{e}}_{\mathrm{J}} &= \mathrm{E}\left\{\left(\begin{bmatrix}\hat{\underline{x}}^{\mathrm{e}}_A \\ \hat{\underline{x}}^{\mathrm{e}}_B\end{bmatrix} - \begin{bmatrix}\mathbf{I} \\ \mathbf{I}\end{bmatrix}\underline{\boldsymbol{x}}\right)\left(\begin{bmatrix}\hat{\underline{x}}^{\mathrm{e}}_A \\ \hat{\underline{x}}^{\mathrm{e}}_B\end{bmatrix} - \begin{bmatrix}\mathbf{I} \\ \mathbf{I}\end{bmatrix}\underline{\boldsymbol{x}}\right)^{\mathrm{T}}\right\} \\ &= \begin{bmatrix}\mathbf{C}^{\mathrm{e}}_A & \mathbf{C}^{\mathrm{e}}_{AB} \\ \mathbf{C}^{\mathrm{e}}_{BA} & \mathbf{C}^{\mathrm{e}}_A\end{bmatrix} \ .\end{aligned}$$

This relationship implies that $\hat{\underline{x}}^{\mathrm{e}}_A$ and $\hat{\underline{x}}^{\mathrm{e}}_B$ are estimates on two "copies" of $\underline{\boldsymbol{x}}$.

1) Prediction: Even if the local estimators are initialized with independent estimates, i.e., $\mathbf{C}^{\mathrm{e}}_{AB} = \mathbf{0}$, it is the common process noise that enters through the *prediction step* and causes non-zero cross-correlations. Given a process model

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

for time step k to $k+1$, the process model related to the joint estimate is

$$\begin{bmatrix}\underline{\boldsymbol{x}}_{k+1} \\ \underline{\boldsymbol{x}}_{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k\end{bmatrix}\begin{bmatrix}\underline{\boldsymbol{x}}_k \\ \underline{\boldsymbol{x}}_k\end{bmatrix} + \begin{bmatrix}\mathbf{I} \\ \mathbf{I}\end{bmatrix}\underline{\boldsymbol{w}}_k \ , \tag{2}$$

where the process noise $\underline{\boldsymbol{w}}_k$ affects both "copies" in the same way. Accordingly, the time update of the joint covariance matrix reads

$$\mathbf{C}^{\mathrm{p}}_{\mathrm{J}} = \begin{bmatrix}\mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k\end{bmatrix}\mathbf{C}^{\mathrm{e}}_{\mathrm{J}}\begin{bmatrix}\mathbf{A}^{\mathrm{T}}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{\mathrm{T}}_k\end{bmatrix} + \begin{bmatrix}\mathbf{C}^{\boldsymbol{w}}_k & \mathbf{C}^{\boldsymbol{w}}_k \\ \mathbf{C}^{\boldsymbol{w}}_k & \mathbf{C}^{\boldsymbol{w}}_k\end{bmatrix} \ ,$$

where $\mathbf{C}^{\boldsymbol{w}}_k$ is the covariance matrix of $\underline{\boldsymbol{w}}_k$. Hence, the predicted cross-covariance matrix becomes

$$\mathbf{C}^{\mathrm{p}}_{AB} = \mathbf{A}_k\, \mathbf{C}^{\mathrm{e}}_{AB}\, \mathbf{A}^{\mathrm{T}}_k + \mathbf{C}^{\boldsymbol{w}}_k \ .$$

2) Global Filtering: The *filtering step* in a multi-sensor system usually benefits from the conditional independence of the measurements given the current state. However, due to cross-correlations, the estimation parameters $(\hat{\underline{x}}^{\mathrm{p}}_B, \mathbf{C}^{\mathrm{p}}_B)$ do not remain unaffected by a measurement update at node A, i.e., the optimal processing of a measurement updates the entire joint state estimate. At a given time step k, let $(\hat{\underline{x}}^{\mathrm{p}}_{\mathrm{J}}, \mathbf{C}^{\mathrm{p}}_{\mathrm{J}})$ be the prior joint estimate to be updated by measurement information. A measurement $\hat{\underline{z}}_A$ at node A is related to the joint state space by the sensor model

$$\underline{\boldsymbol{z}}_A = \mathbf{H}_{\mathrm{J}}\begin{bmatrix}\underline{\boldsymbol{x}} \\ \underline{\boldsymbol{x}}\end{bmatrix} + \underline{\boldsymbol{v}}_A \tag{3}$$

with $\mathbf{H}_{\mathrm{J}} = [\mathbf{H}_A\ \mathbf{0}]$ and measurement noise $\underline{\boldsymbol{v}}_A$. The Kalman gain [7] then yields

$$\begin{aligned}\mathbf{K}_{\mathrm{J}} &= \mathbf{C}^{\mathrm{p}}_{\mathrm{J}}\mathbf{H}^{\mathrm{T}}_{\mathrm{J}}\left(\mathbf{H}_{\mathrm{J}}\mathbf{C}^{\mathrm{p}}_{\mathrm{J}}\mathbf{H}^{\mathrm{T}}_{\mathrm{J}} + \mathbf{C}^{\boldsymbol{v}}_A\right)^{-1} \\ &= \begin{bmatrix}\mathbf{C}^{\mathrm{p}}_A\mathbf{H}^{\mathrm{T}}_A\left(\mathbf{H}_A\mathbf{C}^{\mathrm{p}}_A\mathbf{H}^{\mathrm{T}}_A + \mathbf{C}^{\boldsymbol{v}}_A\right)^{-1} \\ \mathbf{C}^{\mathrm{p}}_{BA}\mathbf{H}^{\mathrm{T}}_A\left(\mathbf{H}_A\mathbf{C}^{\mathrm{p}}_A\mathbf{H}^{\mathrm{T}}_A + \mathbf{C}^{\boldsymbol{v}}_A\right)^{-1}\end{bmatrix} =: \begin{bmatrix}\mathbf{K}_A \\ \mathbf{K}_{BA}\end{bmatrix}\end{aligned}$$

with non-zero $\mathbf{K}_{BA}$. The joint state estimate and covariance matrix have to be updated according to

$$\begin{bmatrix}\hat{\underline{x}}^{\mathrm{e}}_A \\ \hat{\underline{x}}^{\mathrm{e}}_B\end{bmatrix} = (\mathbf{I} - \mathbf{K}_{\mathrm{J}}\mathbf{H}_{\mathrm{J}})\begin{bmatrix}\hat{\underline{x}}^{\mathrm{p}}_A \\ \hat{\underline{x}}^{\mathrm{p}}_B\end{bmatrix} + \mathbf{K}_{\mathrm{J}}\,\hat{\underline{z}}_A \tag{4}$$

and

$$\mathbf{C}^{\mathrm{e}}_{\mathrm{J}} = (\mathbf{I} - \mathbf{K}_{\mathrm{J}}\mathbf{H}_{\mathrm{J}})\mathbf{C}^{\mathrm{p}}_{\mathrm{J}}(\mathbf{I} - \mathbf{K}_{\mathrm{J}}\mathbf{H}_{\mathrm{J}})^{\mathrm{T}} + \mathbf{K}_{\mathrm{J}}\mathbf{C}^{\boldsymbol{v}}_A\mathbf{K}^{\mathrm{T}}_{\mathrm{J}} \ , \tag{5}$$

respectively.

3) Local Filtering: Apparently, the measurement $\hat{\underline{z}}_A$ also updates $\hat{\underline{x}}^{\mathrm{p}}_B$ in (4), which is the optimal strategy but is not desired if nodes A and B are intended to operate independently. An update of only $\hat{\underline{x}}^{\mathrm{p}}_A$ corresponds to setting $\mathbf{K}_{BA} := \mathbf{0}$, which is implicitly done when distributed track estimates are computed without interchanging measurements. This states the reason why track-to-track fusion only provides suboptimal results. Although $(\hat{\underline{x}}^{\mathrm{p}}_B, \mathbf{C}^{\mathrm{p}}_B)$ is not affected by $\hat{\underline{z}}^{\mathrm{p}}_A$, the cross-covariance matrix $\mathbf{C}^{\mathrm{p}}_{AB} = (\mathbf{C}^{\mathrm{p}}_{BA})^{\mathrm{T}}$ has to be updated. The parameters after the local measurement update at node A are given by

$$\hat{\underline{x}}^{\mathrm{e}}_A = (\mathbf{I} - \mathbf{K}_A\mathbf{H}_A)\hat{\underline{x}}^{\mathrm{p}}_A + \mathbf{K}_A\,\hat{\underline{z}}_A \ , \tag{6}$$

$$\hat{\underline{x}}^{\mathrm{e}}_B = \hat{\underline{x}}^{\mathrm{p}}_B \ , \tag{7}$$

$$\mathbf{C}^{\mathrm{e}}_A = (\mathbf{I} - \mathbf{K}_A\mathbf{H}_A)\mathbf{C}^{\mathrm{p}}_A(\mathbf{I} - \mathbf{K}_A\mathbf{H}_A)^{\mathrm{T}} + \mathbf{K}_A\mathbf{C}^{\boldsymbol{v}}_A\mathbf{K}^{\mathrm{T}}_A \ , \tag{8}$$

$$\mathbf{C}^{\mathrm{e}}_{AB} = (\mathbf{C}^{\mathrm{e}}_{BA})^{\mathrm{T}} = (\mathbf{I} - \mathbf{K}_A\mathbf{H}_A)\mathbf{C}^{\mathrm{p}}_{AB} \ , \tag{9}$$

$$\mathbf{C}^{\mathrm{e}}_B = \mathbf{C}^{\mathrm{p}}_B \ . \tag{10}$$

4) Fusion: The fusion of two track estimates can be achieved by means of the Bar-Shalom/Campo formulas [3]. This fusion rule is an optimal linear combination

$$\hat{\underline{x}}_{\text{fus}} = (\mathbf{I} - \mathbf{K}_{\text{fus}})\,\hat{\underline{x}}_A^{\text{e}} + \mathbf{K}_{\text{fus}}\,\hat{\underline{x}}_B^{\text{e}} \tag{11}$$

that minimizes the MSE $\mathrm{E}\left\{(\hat{\underline{x}}_{\text{fus}} - \underline{\boldsymbol{x}})^{\mathrm{T}}(\hat{\underline{x}}_{\text{fus}} - \underline{\boldsymbol{x}})\right\}$. The required gain is

$$\mathbf{K}_{\text{fus}} = (\mathbf{C}_A - \mathbf{C}_{AB}) \cdot (\mathbf{C}_A + \mathbf{C}_B - \mathbf{C}_{AB} - \mathbf{C}_{BA})^{-1}$$

and yields the error covariance matrix

$$\mathbf{C}_{\text{fus}} = \mathbf{C}_A - \mathbf{K}_{\text{fus}}\,(\mathbf{C}_A - \mathbf{C}_{BA}) \ . \tag{12}$$

This combination rule provides maximum likelihood fusion results [4]. In particular, with local processing and subsequent fusion of estimates, we cannot obtain the same results as with a centralized Kalman filter, which has access to all measurements and computes an MMSE estimate.

IV. Linear Track-to-track Fusion with Gaussian Densities

For the later analysis of Gaussian mixtures in Sec. V, a characterization of dependencies in terms of joint probability densities is required. Therefore, the considerations in the previous Sec. III are viewed from a different perspective.

In Bayesian filtering theory, conditional probability densities for the state $\underline{\boldsymbol{x}}$ are computed [19]. In the special case of linear process and sensor models, as in the preceding section, and normally distributed noise terms in (1) and (3), Kalman filtering is related to the processing of the underlying Gaussian probability densities. This relationship can be established because the covariance matrix (5) does not depend on the actual measurements (see, e.g., [20]). The prediction step and the filtering step correspond to the Chapman-Kolmogorov integral and Bayes' rule for probability densities, respectively. At a given time step k, dependencies between the estimated densities $f^{\text{e}}(\underline{x}_A) = \mathcal{N}(\underline{x}_A; \hat{\underline{x}}_A^{\text{e}}, \mathbf{C}_A^{\text{e}})$ and $f^{\text{e}}(\underline{x}_B) = \mathcal{N}(\underline{x}_B; \hat{\underline{x}}_B^{\text{e}}, \mathbf{C}_B^{\text{e}})$ can again be expressed in terms of a joint density representation

$$f^{\text{e}}(\underline{x}_A, \underline{x}_B) = \mathcal{N}\left(\begin{bmatrix}\underline{x}_A\\ \underline{x}_B\end{bmatrix}; \begin{bmatrix}\hat{\underline{x}}_A^{\text{e}}\\ \hat{\underline{x}}_B^{\text{e}}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^{\text{e}} & \mathbf{C}_{AB}^{\text{e}}\\ \mathbf{C}_{BA}^{\text{e}} & \mathbf{C}_B^{\text{e}}\end{bmatrix}\right) \ , \tag{13}$$

where the superscript e denotes the current estimated density

$$f^{\text{e}}(\underline{x}_k) = f(\underline{x}_k \,|\, \hat{\underline{z}}_k, \cdots, \hat{\underline{z}}_0)$$

and superscript p is used for predicted densities

$$f^{\text{p}}(\underline{x}_{k+1}) = f(\underline{x}_{k+1} \,|\, \hat{\underline{z}}_k, \cdots, \hat{\underline{z}}_0) \ .$$

For a shorter notation, we write $\underline{y} = \underline{x}_{k+1}$ in the following.

1) Prediction: For the prediction of the joint Gaussian density (13) from time step k to $k+1$, the transition density $f(\underline{y}_A, \underline{y}_B \,|\, \underline{x}_A, \underline{x}_B)$ with $\underline{y} := \underline{x}_{k+1}$ has to be derived from the process model (2), which is then given by

$$\begin{aligned} f(\underline{y}_A, \underline{y}_B \,|\, \underline{x}_A, \underline{x}_B) &= \int \delta(\underline{y}_A - \mathbf{A}_k\,\underline{x}_A - \underline{w}_k) \cdot \\ &\quad \delta(\underline{y}_B - \mathbf{A}_k\,\underline{x}_B - \underline{w}_k) \cdot f(\underline{w}_k)\,\mathrm{d}\underline{w}_k \end{aligned} \tag{14}$$

with $f(\underline{w}_k) = \mathcal{N}(\underline{w}_k; \underline{0}, \mathbf{C}_k^{\boldsymbol{w}})$. Although this function is no valid transition density due to the product of dirac deltas, the Chapman-Kolmogorov integral for the prediction becomes

$$\begin{aligned} f^{\text{p}}(\underline{y}_A, \underline{y}_B) &= \int\!\!\int f(\underline{y}_A, \underline{y}_B \,|\, \underline{x}_A, \underline{x}_B) \\ &\qquad \cdot f^{\text{e}}(\underline{x}_A, \underline{x}_B)\,\mathrm{d}\underline{x}_A\,\mathrm{d}\underline{x}_B \\ &= \int \mathcal{N}\left(\begin{bmatrix}\underline{y}_A - \underline{w}_k\\ \underline{y}_B - \underline{w}_k\end{bmatrix}; \begin{bmatrix}\mathbf{A}_k\hat{\underline{x}}_A^{\text{e}}\\ \mathbf{A}_k\hat{\underline{x}}_B^{\text{e}}\end{bmatrix}, \right. \\ &\qquad \left. \begin{bmatrix}\mathbf{A}_k\mathbf{C}_A^{\text{e}}\mathbf{A}_k^{\mathrm{T}} & \mathbf{A}_k\mathbf{C}_{AB}^{\text{e}}\mathbf{A}_k^{\mathrm{T}}\\ \mathbf{A}_k\mathbf{C}_{BA}^{\text{e}}\mathbf{A}_k^{\mathrm{T}} & \mathbf{A}_k\mathbf{C}_B^{\text{e}}\mathbf{A}_k^{\mathrm{T}}\end{bmatrix}\right) f(\underline{w}_k)\,\mathrm{d}\underline{w}_k \\ &= \mathcal{N}\left(\begin{bmatrix}\underline{y}_A\\ \underline{y}_B\end{bmatrix}; \begin{bmatrix}\hat{\underline{x}}_A^{\text{p}}\\ \hat{\underline{x}}_B^{\text{p}}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^{\text{p}} & \mathbf{C}_{AB}^{\text{p}}\\ \mathbf{C}_{BA}^{\text{p}} & \mathbf{C}_B^{\text{p}}\end{bmatrix}\right) \end{aligned} \tag{15}$$

by rearranging the integrals, exploiting the sifting property, and completing the square in the exponent. As in the previous subsection, the parameters of the predicted density are

$$\begin{aligned} \hat{\underline{x}}_X^{\text{p}} &= \mathbf{A}_k\,\hat{\underline{x}}_X^{\text{e}} \ , \\ \mathbf{C}_Y^{\text{p}} &= \mathbf{A}_k\,\mathbf{C}_Y^{\text{e}}\,\mathbf{A}_k^{\mathrm{T}} + \mathbf{C}_k^{\boldsymbol{w}} \end{aligned}$$

for $X = A, B$ and $Y = A, AB, BA, B$. In contrast to the standard prediction step, the dimension of the process noise term is only a fraction of the dimension of the joint state.

2) Global Filtering: In the filtering step, the likelihood that corresponds to the sensor model (3) and measurement $\hat{\underline{z}}_A$ has to be employed. For $\underline{v}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^{\boldsymbol{v}})$, the likelihood

$$\begin{aligned} f(\hat{\underline{z}}_A \,|\, \underline{x}_A) &= \int \delta(\hat{\underline{z}}_A - \mathbf{H}_A\,\underline{x}_A - \underline{v}_k)\,\mathcal{N}(\underline{v}_k; \underline{0}, \mathbf{C}_k^{\boldsymbol{v}})\,\mathrm{d}\underline{v}_k \\ &= c \cdot \mathcal{N}(\underline{x}_A; \hat{\underline{x}}_A^{L}, \mathbf{C}_A^{L}) \end{aligned} \tag{16}$$

yields an unnormalized Gaussian density, and it can be rewritten as the likelihood

$$f(\hat{\underline{z}}_A \,|\, \underline{x}_A, \underline{x}_B) = \tilde{c} \cdot \mathcal{N}\left(\begin{bmatrix}\underline{x}_A\\ \underline{x}_B\end{bmatrix}; \hat{\underline{x}}_{\text{J}}^{L}; \mathbf{C}_{\text{J}}^{L}\right) \tag{17}$$

for the joint state space, which has the parameters

$$\begin{aligned} (\mathbf{C}_{\text{J}}^{L})^{-1} &= \mathbf{T}^{\mathrm{T}}(\mathbf{C}_A^{L})^{-1}\mathbf{T} \ , \\ (\mathbf{C}_{\text{J}}^{L})^{-1}\hat{\underline{x}}_{\text{J}}^{L} &= \mathbf{T}^{\mathrm{T}}(\mathbf{C}_A^{L})^{-1}\hat{\underline{x}}_A^{L} \end{aligned}$$

given in the information form [21] with $\mathbf{T} = [\mathbf{I}\ \mathbf{0}]$. The measurement update can now be carried out by means of Bayes' theorem for probability densities, i.e.,

$$\begin{aligned} f^{\text{e}}(\underline{x}_A, \underline{x}_B) &= C \cdot f^{\text{p}}(\underline{x}_A, \underline{x}_B) \cdot f(\hat{\underline{z}}_A \,|\, \underline{x}_A, \underline{x}_B) \\ &= \mathcal{N}\left(\begin{bmatrix}\underline{x}_A\\ \underline{x}_B\end{bmatrix}; \hat{\underline{x}}_{\text{J}}^{\text{e}}; \mathbf{C}_{\text{J}}^{\text{e}}\right) \ , \end{aligned} \tag{18}$$

which is the product of two Gaussian functions with normalization constant C. The parameters $\hat{\underline{x}}_{\text{J}}^{\text{e}}$ and $\mathbf{C}_{\text{J}}^{\text{e}}$ comply with the Kalman filter formulas (4) and (5).

3) Local Filtering: In distributed track-to-track fusion applications, a measurement $\hat{\underline{z}}_A$ usually implies that only the marginal density $f^{\text{p}}(\underline{x}_A)$ is to be updated in place of the global update (18) for $f^{\text{p}}(\underline{x}_A, \underline{x}_B)$. However, the reformulation of the local likelihood (16) into the global likelihood (17) reveals that also the marginal density $f^{\text{p}}(\underline{x}_B)$ is affected by a measurement at sensor node A. Hence, a local filtering step at node A without affecting the estimated density at node B cannot simply be expressed in terms of Bayes' theorem applied to the joint density, as in (18). In order to compute the joint density after the local update, we first consider the linear transformation $(\mathbf{I}-\mathbf{K}_A\mathbf{H}_A)\underline{x}_A$ in (6), which yields the joint density

$$
\begin{aligned}
f^{\mathrm{e},1}(\underline{x}_A, \underline{x}_B) &= \int \delta(\underline{x}_A - \mathbf{K}_1\,\xi) f^{\mathrm{p}}(\xi, \underline{x}_B)\,\mathrm{d}\xi \\
&= c f^{\mathrm{p}}(\mathbf{K}_1^{-1}\underline{x}_A, \underline{x}_B) \\
&= c\mathcal{N}\left(\begin{bmatrix}\mathbf{K}_1^{-1}\underline{x}_A \\ \underline{x}_B\end{bmatrix}; \begin{bmatrix}(\mathbf{K}_1^{-1}\mathbf{K}_1)\hat{\underline{x}}_A \\ \hat{\underline{x}}_B\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^{\mathrm{e}} & \mathbf{C}_{AB}^{\mathrm{e}} \\ \mathbf{C}_{BA}^{\mathrm{e}} & \mathbf{C}_B^{\mathrm{e}}\end{bmatrix}\right) \\
&= \mathcal{N}\left(\begin{bmatrix}\underline{x}_A \\ \underline{x}_B\end{bmatrix}; \begin{bmatrix}\mathbf{K}_1\hat{\underline{x}}_A \\ \hat{\underline{x}}_B\end{bmatrix}, \begin{bmatrix}\mathbf{K}_1\mathbf{C}_A^{\mathrm{e}}\mathbf{K}_1^{\mathrm{T}} & \mathbf{K}_1\mathbf{C}_{AB}^{\mathrm{e}} \\ \mathbf{C}_{BA}^{\mathrm{e}}\mathbf{K}_1^{\mathrm{T}} & \mathbf{C}_B^{\mathrm{e}}\end{bmatrix}\right)
\end{aligned} \tag{19}
$$

with $\mathbf{K}_1 = (\mathbf{I}-\mathbf{K}_A\mathbf{H}_A)$. The sum in (6) can then be expressed in terms of the convolution

$$
\begin{aligned}
f^{\mathrm{e}}(\underline{x}_A, \underline{x}_B) = &\int f^{\mathrm{e},1}(\underline{x}_A - \underline{v}, \underline{x}_B)\cdot \\
&\quad \mathcal{N}(\underline{v}; \mathbf{K}_A\,\hat{\underline{z}}_A, \mathbf{K}_A\mathbf{C}_A^{\boldsymbol{v}}\mathbf{K}_A^{\mathrm{T}})\,\mathrm{d}\underline{v} \\
= \mathcal{N}\Bigg(&\begin{bmatrix}\underline{x}_A \\ \underline{x}_B\end{bmatrix}; \begin{bmatrix}\mathbf{K}_1\hat{\underline{x}}_A + \mathbf{K}_A\,\hat{\underline{z}}_A \\ \hat{\underline{x}}_B\end{bmatrix}, \\
&\begin{bmatrix}\mathbf{K}_1\mathbf{C}_A^{\mathrm{e}}\mathbf{K}_1^{\mathrm{T}} + \mathbf{K}_A\mathbf{C}_A^{\boldsymbol{v}}\mathbf{K}_A^{\mathrm{T}} & \mathbf{K}_1\mathbf{C}_{AB}^{\mathrm{e}} \\ \mathbf{C}_{BA}^{\mathrm{e}}\mathbf{K}_1^{\mathrm{T}} & \mathbf{C}_B^{\mathrm{e}}\end{bmatrix}\Bigg)\ .
\end{aligned} \tag{20}
$$

The resulting density $f^{\mathrm{e}}(\underline{x}_A, \underline{x}_B)$ has the parameters given in equations (6)-(10).

4) Fusion: For the purpose of fusing two marginal densities $f^{\mathrm{e}}(\underline{x}_A)$ and $f^{\mathrm{e}}(\underline{x}_B)$, the joint density is considered as a likelihood for the state $\underline{\boldsymbol{x}}$, i.e.,

$$
f_{\mathrm{fus}}(\underline{x}) = C\cdot\mathcal{N}\left(\begin{bmatrix}\mathbf{I} \\ \mathbf{I}\end{bmatrix}\underline{x}; \begin{bmatrix}\hat{\underline{x}}_A^{\mathrm{e}} \\ \hat{\underline{x}}_B^{\mathrm{e}}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^{\mathrm{e}} & \mathbf{C}_{AB}^{\mathrm{e}} \\ \mathbf{C}_{BA}^{\mathrm{e}} & \mathbf{C}_B^{\mathrm{e}}\end{bmatrix}\right)\ . \tag{21}
$$

The resulting function is again a Gaussian density. The parameters can be derived by considering the exponent, using the block inverse of the joint covariance matrix, and multiplying out the brackets:

$$
\begin{aligned}
&-\tfrac{1}{2}\underline{x}^{\mathrm{T}}\begin{bmatrix}\mathbf{I} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{C}_A^{\mathrm{e}} & \mathbf{C}_{AB}^{\mathrm{e}} \\ \mathbf{C}_{BA}^{\mathrm{e}} & \mathbf{C}_B^{\mathrm{e}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{I} \\ \mathbf{I}\end{bmatrix}\underline{x} + \dots \\
&= -\tfrac{1}{2}\underline{x}^{\mathrm{T}}\Big(\mathbf{C}_A^{-1} + \mathbf{C}_A^{-1}\mathbf{C}_{AB}\big(\mathbf{C}_B - \mathbf{C}_{BA}\mathbf{C}_A^{-1}\mathbf{C}_{AB}\big)^{-1}\mathbf{C}_{BA}\mathbf{C}_A^{-1} \\
&\qquad - \mathbf{C}_A^{-1}\mathbf{C}_{AB}\big(\mathbf{C}_B - \mathbf{C}_{BA}\mathbf{C}_A^{-1}\mathbf{C}_{AB}\big)^{-1} \\
&\qquad - \big(\mathbf{C}_B - \mathbf{C}_{BA}\mathbf{C}_A^{-1}\mathbf{C}_{AB}\big)^{-1}\mathbf{C}_{BA}\mathbf{C}_A^{-1} \\
&\qquad + \big(\mathbf{C}_B - \mathbf{C}_{BA}\mathbf{C}_A^{-1}\mathbf{C}_{AB}\big)^{-1}\Big)\underline{x} + \dots \\
&= -\tfrac{1}{2}\underline{x}^{\mathrm{T}}\Big(\mathbf{C}_A^{-1} + \mathbf{C}_A^{-1}(\mathbf{C}_A - \mathbf{C}_{AB})\cdot \\
&\qquad \big(\mathbf{C}_B - \mathbf{C}_{BA}\mathbf{C}_A^{-1}\mathbf{C}_{AB}\big)^{-1}(\mathbf{C}_A - \mathbf{C}_{BA})\mathbf{C}_A^{-1}\Big)\underline{x} + \dots \\
&= -\tfrac{1}{2}\underline{x}^{\mathrm{T}}\Big(\mathbf{C}_A - (\mathbf{C}_A - \mathbf{C}_{AB})\cdot \\
&\qquad \big(\mathbf{C}_A + \mathbf{C}_B - \mathbf{C}_{AB} - \mathbf{C}_{BA}\big)^{-1}(\mathbf{C}_A - \mathbf{C}_{BA})\Big)^{-1}\underline{x} + \dots\ ,
\end{aligned} \tag{22}
$$

where the last equation can be derived by applying the Woodbury matrix identity. As expected, the resulting covariance matrix is identical to (12), and further rearrangements will lead to the mean (11). The fusion rule (21) implies that the resulting density lies on the "diagonal" of the joint density [15], [16], as illustrated in Figure 1(a) for a more complex density.

On the basis of these considerations, we are now able to generalize the joint density representation and fusion of track estimates to a special class of nonlinear estimation problems.

V. Nonlinear Track-to-track Fusion with Gaussian Mixture Densities

Nonlinear state estimation is in itself a formidable task but becomes even more complicated when estimates have to be computed in a distributed fashion. As in the linear case, optimal fusion of local tracking results requires knowledge about the underlying dependencies. Again, the dependency structure is linked with the joint state space representation of the estimates to be fused, which requires the consideration of far higher dimensional probability densities than in the centralized case, i.e., the joint state representation has a multiple of the dimension of the original state space.

Gaussian mixture densities form the basis for many efficient filtering algorithms [22] to tackle nonlinear estimation problems. In target tracking applications, they can be used to model the uncertainty about a target's maneuver [23] or to represent uncertainties in multi-target tracking scenarios [17]. Against the background of distributed multi-sensor systems, the problem of fusing Gaussian mixture estimates has been studied in [11], [12], [24], where the treatment of common information or conservative fusion rules are considered. In [25], [26], parameterized joint densities for Gaussian mixture estimates have been derived with the objective of providing a generalized correlation coefficient for Gaussian mixtures. This section complements these studies with a systematic derivation of the dependencies between two Gaussian mixture estimates.

We commence this section with the construction of the joint density of two independent Gaussian mixture marginals. The local Gaussian mixture densities

$$
f^{\mathrm{e}}(\underline{x}_A) = \sum_{i=1}^{N_A}\omega_A^i\mathcal{N}\left(\underline{x}_A; \hat{\underline{x}}_A^i, \mathbf{C}_A^i\right) \tag{23}
$$

and

$$
f^{\mathrm{e}}(\underline{x}_B) = \sum_{i=1}^{N_B}\omega_B^i\mathcal{N}\left(\underline{x}_B; \hat{\underline{x}}_B^i, \mathbf{C}_B^i\right) \tag{24}
$$

at sensor nodes A and B, respectively, are assumed to be independent. In this case, the estimates can be fused by multiplication and renormalization [15], [16], which yields

$$
\begin{aligned}
f^{\mathrm{fus}}(\underline{x}) &= C\sum_{i=1}^{N_A}\sum_{i=1}^{N_B}\omega_A^i\omega_B^j\mathcal{N}\left(\underline{x}; \hat{\underline{x}}_A^i, \mathbf{C}_A^i\right)\cdot\mathcal{N}\left(\underline{x}; \hat{\underline{x}}_B^j, \mathbf{C}_B^j\right) \\
&= C\sum_{i=1}^{N_A}\sum_{i=1}^{N_B}\omega_A^i\omega_B^j\mathcal{N}\left(\begin{bmatrix}\underline{x} \\ \underline{x}\end{bmatrix}; \begin{bmatrix}\hat{\underline{x}}_A^i \\ \hat{\underline{x}}_B^j\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^i & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_B^j\end{bmatrix}\right)\ .
\end{aligned} \tag{25}
$$

Evidently, the last expression leads us to the joint density

$$
f^{\mathrm{J}}(\underline{x}_A, \underline{x}_B) = \sum_{i=1}^{N_A}\sum_{i=1}^{N_B}\omega_A^i\omega_B^j\mathcal{N}\left(\begin{bmatrix}\underline{x}_A \\ \underline{x}_B\end{bmatrix}; \begin{bmatrix}\hat{\underline{x}}_A^i \\ \hat{\underline{x}}_B^j\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^i & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_B^j\end{bmatrix}\right)
$$

of $f^{\mathrm{e}}(\underline{x}_A)$ and $f^{\mathrm{e}}(\underline{x}_B)$, which has $N_A \cdot N_B$ components. The following example conveys an impression of the joint dependency structure of Gaussian mixtures.

Example: Joint Gaussian Mixture Density

The dependency structure, i.e., the joint density, of the independent Gaussian mixtures

$$
\begin{aligned}
f(x_A) &= 0.3\mathcal{N}(x_A; -0.5, 0.5) + 0.7\mathcal{N}(x_A; 3.5, 1.5) \\
f(x_B) &= 0.7\mathcal{N}(x_B; 1, 1.2) + 0.3\mathcal{N}(x_B; 5, 0.8)
\end{aligned}
$$

is given by a two-dimensional joint Gaussian mixture with four components

$$0.21\,\mathcal{N}\left(x,\begin{bmatrix}-0.5\\1\end{bmatrix},\begin{bmatrix}0.5 & 0\\0 & 1.2\end{bmatrix}\right),0.49\,\mathcal{N}\left(x,\begin{bmatrix}3.5\\1\end{bmatrix},\begin{bmatrix}1.5 & 0\\0 & 1.2\end{bmatrix}\right),$$

$$0.09\,\mathcal{N}\left(x,\begin{bmatrix}-0.5\\5\end{bmatrix},\begin{bmatrix}0.5 & 0\\0 & 0.8\end{bmatrix}\right),0.21\,\mathcal{N}\left(x,\begin{bmatrix}3.5\\5\end{bmatrix},\begin{bmatrix}1.5 & 0\\0 & 0.8\end{bmatrix}\right).$$

The joint density is shown in Fig. 1(a). The fusion result is plotted in Fig. 1(b) and lies on the "diagonal" of the joint density.

1) Prediction: As in the linear case, the prediction step states the reason for dependent local estimates. Let the transition from a time step k to $k+1$ be given by the process model

$$\underline{\boldsymbol{y}} = \mathbf{A}\,\underline{\boldsymbol{x}} + \underline{\boldsymbol{w}}\ , \tag{26}$$

where the density of $\underline{\boldsymbol{w}}$ now is a Gaussian mixture

$$f(\underline{w}) = \sum_{i=1}^{N^w} \omega^{w,i}\,\mathcal{N}(\underline{w};\hat{\underline{w}}^i,\mathbf{C}^{w,i})\ . \tag{27}$$

In order to perform the time update of the current estimated joint Gaussian mixture

$$f^{\mathrm{e}}(\underline{y}_A,\underline{y}_B) = \sum_{i=1}^{N_\mathrm{J}} \omega_\mathrm{J}^{\mathrm{e},i}\,\mathcal{N}\left(\begin{bmatrix}\underline{y}_A\\\underline{y}_B\end{bmatrix};\begin{bmatrix}\hat{\underline{x}}_A^{\mathrm{e},i}\\\hat{\underline{x}}_B^{\mathrm{e},i}\end{bmatrix},\begin{bmatrix}\mathbf{C}_A^{\mathrm{e},i} & \mathbf{C}_{AB}^{\mathrm{e},i}\\\mathbf{C}_{BA}^{\mathrm{e},i} & \mathbf{C}_B^{\mathrm{e},i}\end{bmatrix}\right),$$

the Chapman-Kolmogorov integral (15) from Sec. IV is applied, which results into

$$\begin{aligned}f^{\mathrm{p}}(\underline{y}_A,\underline{y}_B) &= \int\int f(\underline{y}_A,\underline{y}_B\,|\,\underline{x}_A,\underline{x}_B)\\ &\qquad\cdot f^{\mathrm{e}}(\underline{x}_A,\underline{x}_B)\,\mathrm{d}\underline{x}_A\,\mathrm{d}\underline{x}_B \\ &= \sum_{i=1}^{N_\mathrm{J}}\sum_{j=1}^{N^w}\int \omega_\mathrm{J}^{\mathrm{e},i}\,\omega^{w,j}\,\mathcal{N}(\underline{w};\hat{\underline{w}}^j,\mathbf{C}^{w,j})\\ &\quad\mathcal{N}\left(\begin{bmatrix}\underline{y}_A-\underline{w}_k\\\underline{y}_B-\underline{w}_k\end{bmatrix};\begin{bmatrix}\mathbf{A}_k\hat{\underline{x}}_A^{\mathrm{e},i}\\\mathbf{A}_k\hat{\underline{x}}_B^{\mathrm{e},i}\end{bmatrix},\begin{bmatrix}\mathbf{A}_k\mathbf{C}_A^{\mathrm{e},i}\mathbf{A}_k^\mathrm{T} & \mathbf{A}_k\mathbf{C}_{AB}^{\mathrm{e},i}\mathbf{A}_k^\mathrm{T}\\\mathbf{A}_k\mathbf{C}_{BA}^{\mathrm{e},i}\mathbf{A}_k^\mathrm{T} & \mathbf{A}_k\mathbf{C}_B^{\mathrm{e},i}\mathbf{A}_k^\mathrm{T}\end{bmatrix}\right)\mathrm{d}\underline{w}_k\\ &= \sum_{i=1}^{N_\mathrm{J}}\sum_{j=1}^{N^w}\omega_\mathrm{J}^{\mathrm{p},i,j}\mathcal{N}\left(\begin{bmatrix}\underline{y}_A\\\underline{y}_B\end{bmatrix};\begin{bmatrix}\hat{\underline{x}}_A^{\mathrm{p},i,j}\\\hat{\underline{x}}_B^{\mathrm{p},i,j}\end{bmatrix},\begin{bmatrix}\mathbf{C}_A^{\mathrm{p},i,j} & \mathbf{C}_{AB}^{\mathrm{p},i,j}\\\mathbf{C}_{BA}^{\mathrm{p},i,j} & \mathbf{C}_B^{\mathrm{p},i,j}\end{bmatrix}\right)\ .\end{aligned} \tag{28}$$

The predicted Gaussian mixture has the parameters

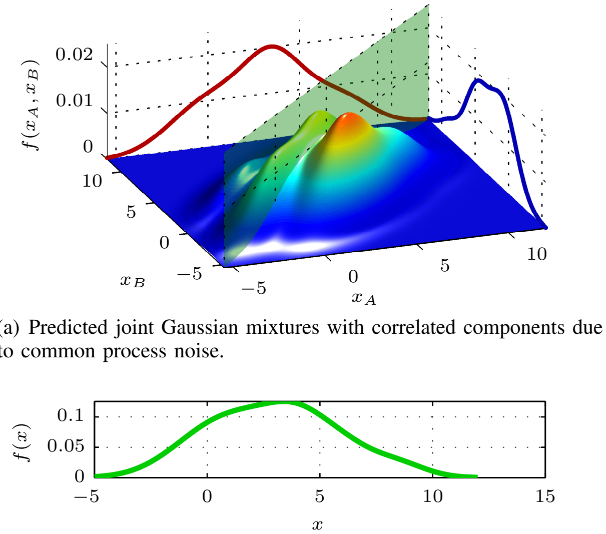
$$\begin{aligned}\omega_\mathrm{J}^{\mathrm{p},i,j} &= \omega_\mathrm{J}^{\mathrm{e},i}\,\omega^{w,j}\ ,\\ \hat{\underline{x}}_X^{\mathrm{p},i,j} &= \mathbf{A}\,\hat{\underline{x}}_X^{\mathrm{e},i}\ ,\\ \mathbf{C}_Y^{\mathrm{p},i,j} &= \mathbf{A}\mathbf{C}_Y^{\mathrm{e},i}\mathbf{A}^\mathrm{T} + \mathbf{C}^{\boldsymbol{w},j}\end{aligned} \tag{29}$$

for $i = 1,\ldots,N_\mathrm{J}, j = 1,\ldots,N^w$ with $X = A,B$ and $Y = A,AB,BA,B$. An example of the prediction step is discussed in the following paragraph.

Example: Prediction of Joint Gaussian Mixture Density
The joint Gaussian mixture depicted in Fig. 1(a) undergoes a prediction step according to the model (26) with $\mathbf{A} = 1$ and the process noise density

$$f(w) = 0.8\,\mathcal{N}(w;0,3) + 0.2\,\mathcal{N}(w;4,1) \tag{30}$$

The prediction result is shown in Fig. 2(a) and possesses $N_\mathrm{J}\cdot N^w = 8$ components. Each component is a Gaussian with a cross-covariance of either $C_{AB} = 3$ or $C_{AB} = 1$, which states the reason for the diagonal orientation of the joint density. The fusion result is shown in Fig. 2(b).



(a) Predicted joint Gaussian mixtures with correlated components due to common process noise.

(b) Fusion result corresponds to the plane.

Figure 2. Predicted joint Gaussian mixtures and fusion result.

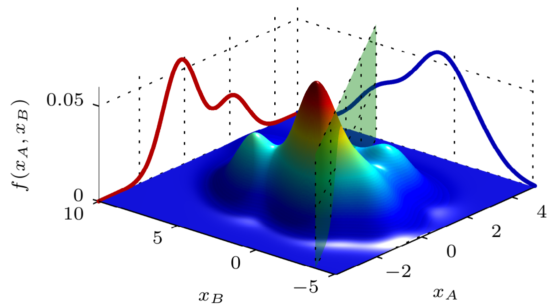
Instead of the linear model (26) with a Gaussian mixture noise terms, other models can be used. Only the transition density (14) has to turn into a Gaussian mixture such that the joint Gaussian mixture representation is preserved. A particularly interesting result of the joint prediction step is that the joint process noise density and hence the joint transition density has the same number of components as in the original state space, i.e., the joint transition density also has N^w components and not, as one might expect, $(N^w)^2$ components.

Linear distributed estimation requires bookkeeping of the cross-covariance matrices $\mathbf{C}_{AB}$. For Gaussian mixtures, the required effort to keep track of the dependency structure between two estimates becomes tremendous: Not only bookkeeping of the cross-covariance matrices $\mathbf{C}_{AB}^i$ for each component is necessary, but also the weighting factors have to be computed correctly. In this regard, the problem of double-counting becomes apparent, which is a well-known problem in distributed estimation systems. The local prediction of (23) and (24) implies that each $\omega^{w,j}$ is multiplied with each weight ω_A^i and each ω_B^i, but (29) reveals that $\omega^{w,j}$ is only incorporated once in the joint prediction result, i.e., $\omega_\mathrm{J}^{\mathrm{p},i,j} = \omega_\mathrm{J}^{\mathrm{e},i}\,\omega^{w,j} = \omega_A^i\,\omega_B^i\,\omega^{w,j}$. As a consequence, the weighting factors cannot simply be stored and managed locally, which proves the high complexity of nonlinear track-to-track fusion.

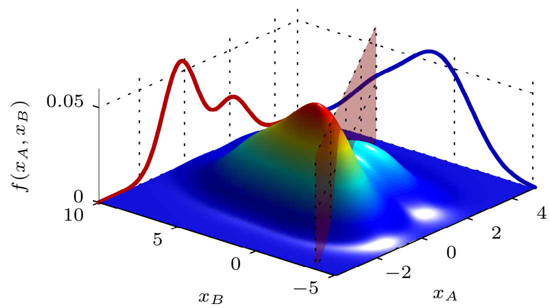
2) Global Filtering: Again w.l.o.g., we consider a measurement at the first state component x_A. The likelihood may refer to an arbitrary nonlinear sensor model but has to be represented by a Gaussian mixture

$$f(\hat{\underline{z}}_k|\underline{x}_A) = c\sum_{i=1}^{N^L}\omega^{L,i}\mathcal{N}\left(\underline{x}_A;\hat{\underline{x}}_A^{L,i},\mathbf{C}_A^{L,i}\right)\ .$$

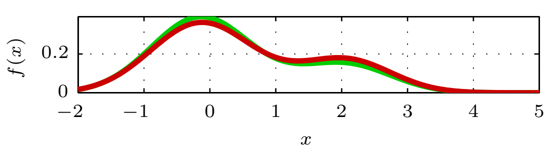
We take the same path as in Sec. IV and, analogously to (17), this likelihood can be seen as a likelihood for the joint state representation. According to Bayes' theorem, the filtering result



(a) Measurement update of red Gaussian mixture density results into global update of joint density.



(b) Only local update of the red Gaussian mixture. The variances of the blue local density are not updated.



(c) Fusion results for both joint densities, which correspond to the green and red plane.

Figure 3. Joint densities after measurement at sensor node A. Global and local update of joint Gaussian mixture.

is a product

$$f^{\mathrm{e}}(\underline{x}_A, \underline{x}_B) = C \cdot f^{\mathrm{p}}(\underline{x}_A, \underline{x}_B) \cdot f(\underline{\hat{z}}_A \,|\, \underline{x}_A, \underline{x}_B) \qquad (31)$$

of Gaussian mixtures, which requires a componentwise computation according to (18) and has $N_{\mathrm{J}} \cdot N^L$ components. The components have the parameters

$$\begin{aligned}
\mathbf{C}_{\mathrm{J}}^{\mathrm{e},i,j} &= \left((\mathbf{C}_{\mathrm{J}}^{\mathrm{p},i})^{-1} + \mathbf{T}^{\mathrm{T}}(\mathbf{C}_A^{L,j})^{-1}\mathbf{T}\right)^{-1} ,\\
\underline{\hat{x}}_{\mathrm{J}}^{\mathrm{e},i,j} &= \mathbf{C}_{\mathrm{J}}^{\mathrm{e},i,j}\left((\mathbf{C}_{\mathrm{J}}^{\mathrm{p},i})^{-1}\underline{x}_A^{\mathrm{p},i} + \mathbf{T}^{\mathrm{T}}(\mathbf{C}_A^{L,j})^{-1}\underline{\hat{x}}_A^{L,i}\right)^{-1} ,\\
\omega_{\mathrm{J}}^{\mathrm{e},i,j} &= C \cdot \omega_{\mathrm{J}}^{\mathrm{p},i} \cdot \omega^{L,j} \cdot \mathcal{N}\left(\underline{\hat{x}}_A^{L,j} - \mathbf{T}\underline{\hat{x}}_{\mathrm{J}}^{\mathrm{p},i}; \underline{0}, \mathbf{T}\mathbf{C}_{\mathrm{J}}^{\mathrm{p},i}\mathbf{T}^{\mathrm{T}} + \mathbf{C}_A^{L,j}\right).
\end{aligned}$$

for $i = 1, \ldots, N_{\mathrm{J}}$, $j = 1, \ldots, N^L$ and $\mathbf{T} = [\mathbf{I}\ \mathbf{0}]$. The first two formulas correspond to the information form of the Kalman filter [21], i.e., mean and covariance parameters are updated by means of Kalman filter formulas. The third equation reveals that the measurement update of a Gaussian mixture requires a measurement-dependent reweighting [22]. The global filtering step is illustrated in the following example.

Example: Global Measurement Update
The predicted joint density from Fig. 2(a) is processed with the likelihood function

$$f(z_A \,|\, x_A, x_B) = 0.8\,\mathcal{N}\big(x_A, -0.3, \tfrac{5}{8}\big) + 0.2\,\mathcal{N}\big(x_A, 2, \tfrac{1}{2}\big)$$

through Bayes' rule (31). The filtering result is shown in Fig. 3(a). Apparently, the density is also altered along state component $\underline{x}_B$.

3) Local Filtering: Sec. IV has revealed that a local measurement update without affecting other marginal estimates cannot be expressed in terms of a Bayesian update step, i.e., the update (31) automatically affects $f^{\mathrm{p}}(\underline{x}_B)$. In order to achieve a local update of $f^{\mathrm{p}}(\underline{x}_A)$, we reduce the update step to

$$f^{\mathrm{e}}(\underline{x}_A) = C \cdot f^{\mathrm{p}}(\underline{x}_A) \cdot f(\underline{\hat{z}}_A \,|\, \underline{x}_A) \ , \qquad (32)$$

which yields the parameters

$$\begin{aligned}
\mathbf{C}_A^{\mathrm{e},i,j} &= \left((\mathbf{C}_A^{\mathrm{p},i})^{-1} + (\mathbf{C}_A^{L,j})^{-1}\right)^{-1} ,\\
\underline{\hat{x}}_A^{\mathrm{e},i,j} &= \mathbf{C}_A^{\mathrm{e},i,j}\left((\mathbf{C}_A^{\mathrm{p},i})^{-1}\underline{x}_A^{\mathrm{p},i} + (\mathbf{C}_A^{L,j})^{-1}\underline{\hat{x}}_A^{L,i}\right)^{-1} ,\\
\omega_A^{\mathrm{e},i,j} &= C \cdot \omega_A^{\mathrm{p},i} \cdot \omega^{L,j} \cdot \mathcal{N}\left(\underline{\hat{x}}_A^{L,j} - \underline{\hat{x}}_A^{\mathrm{p},i}; \underline{0}, \mathbf{C}_A^{\mathrm{p},i} + \mathbf{C}_A^{L,j}\right) .
\end{aligned}$$

After the local update, the dependency structure has to be adapted by applying the calculations (19) and (20) to each component. The cross-covariance matrices in each component are

$$\begin{aligned}
\mathbf{C}_{AB}^{\mathrm{e},i,j} &= (\mathbf{C}_{BA}^{\mathrm{e},i,j})^{\mathrm{T}} = (\mathbf{I} - \mathbf{K}_A^{\mathrm{e},i,j})\,\mathbf{C}_{AB}^{\mathrm{p},i,j} \quad \text{with}\\
\mathbf{K}_A^{\mathrm{e},i,j} &= \mathbf{C}_A^{\mathrm{p},i}\left(\mathbf{C}_A^{\mathrm{p},i} + \mathbf{C}_A^{L,j}\right)^{-1} .
\end{aligned}$$

Although $\mathbf{C}_B$ has not directly been modified, the joint density now has $N_{\mathrm{J}} \cdot N^L$ components, which implies

$$\begin{aligned}
\mathbf{C}_B^{\mathrm{e},i,j_1} &= \mathbf{C}_B^{\mathrm{e},i,j_2} = \mathbf{C}_B^{\mathrm{p},i} ,\\
\underline{\hat{x}}_B^{\mathrm{e},i,j_1} &= \underline{\hat{x}}_B^{\mathrm{e},i,j_2} = \underline{\hat{x}}_B^{\mathrm{p},i} , \quad \text{for } j_1, j_2 \in \{1, \ldots, N^L\} .
\end{aligned}$$

As for the prediction step, an open question is how to manage and where to store the weighting factors. However, the good news is that the updated joint weights $\omega_{\mathrm{J}}^{\mathrm{e},i,j}$ only rely on local information due to $\mathbf{T}\underline{\hat{x}}_{\mathrm{J}}^{\mathrm{p},i} = \underline{\hat{x}}_A^{\mathrm{p},i}$ and $\mathbf{T}\mathbf{C}_{\mathrm{J}}^{\mathrm{p},i}\mathbf{T}^{\mathrm{T}} = \mathbf{C}_A^{\mathrm{p},i}$. Hence, the updated weighting factors are identical for global and local filtering, i.e., $\omega_{\mathrm{J}}^{\mathrm{e},i,j} = \omega_A^{\mathrm{e},i,j}$. The difference between local and global filtering can be seen in the following example.

Example: Local Measurement Update
The predicted joint density from Fig. 2(a) is considered, and only the marginal $f^{\mathrm{p}}(x_A)$ is updated with the likelihood

$$f(z_A \,|\, x_A) = 0.8\,\mathcal{N}\big(x_A, -0.3, \tfrac{5}{8}\big) + 0.2\,\mathcal{N}\big(x_A, 2, \tfrac{1}{2}\big) \qquad (33)$$

through Bayes' rule (32). The filtering result is shown in Fig. 3(b). The only parameters that are altered along state component $\underline{x}_B$ are the weighting factors since they depend on the actual measurement.

4) Fusion: The fusion of two locally estimated Gaussian mixtures $f^{\mathrm{e}}(\underline{x}_A)$ and $f^{\mathrm{e}}(\underline{x}_B)$ is again analogous to (21), i.e., the joint density is interpreted as a likelihood and evaluated on its diagonal according to

$$\begin{aligned}
f_{\text{fus}}(\underline{x}) &= C\sum_{i=1}^{N_{\mathrm{J}}} \omega_{\mathrm{J}}^i \mathcal{N}\left(\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix}\underline{x}; \begin{bmatrix}\underline{\hat{x}}_A^{\mathrm{e},i}\\ \underline{\hat{x}}_B^{\mathrm{e},i}\end{bmatrix}, \begin{bmatrix}\mathbf{C}_A^{\mathrm{e},i} & \mathbf{C}_{AB}^{\mathrm{e},i}\\ \mathbf{C}_{BA}^{\mathrm{e},i} & \mathbf{C}_B^{\mathrm{e},i}\end{bmatrix}\right) \qquad (34)\\
&= \sum_{i=1}^{N_{\text{fus}}} \omega_{\text{fus}}^i \,\mathcal{N}\left(\underline{x}; \underline{\hat{x}}_{\text{fus}}^i, \mathbf{C}_{\text{fus}}^i\right) .
\end{aligned}$$

For each exponent, the calculation (22) provides the parameters

$$\begin{aligned}
\underline{\hat{x}}_{\text{fus}}^i &= (\mathbf{I} - \mathbf{K}_{\text{fus}}^i)\,\underline{\hat{x}}_A^{\mathrm{e},i} + \mathbf{K}_{\text{fus}}^i\,\underline{\hat{x}}_B^{\mathrm{e},i} ,\\
\mathbf{C}_{\text{fus}}^i &= \mathbf{C}_A^{\mathrm{e},i} - \mathbf{K}_{\text{fus}}^i\left(\mathbf{C}_A^{\mathrm{e},i} - \mathbf{C}_{BA}^{\mathrm{e},i}\right) ,\\
\omega_{\text{fus}}^i &= C \cdot \omega_{\mathrm{J}}^i \cdot \mathcal{N}\left(\underline{\hat{x}}_A^{\mathrm{e},i} - \underline{\hat{x}}_B^{\mathrm{e},i}; \underline{0}, \mathbf{C}_A^{\mathrm{e},i} + \mathbf{C}_B^{\mathrm{e},i} - \mathbf{C}_{AB}^{\mathrm{e},i} - \mathbf{C}_{BA}^{\mathrm{e},i}\right)
\end{aligned}$$

where the gain is defined by

$$\mathbf{K}^i_{\text{fus}} = (\mathbf{C}^{e,i}_A - \mathbf{C}^{e,i}_{AB}) \cdot \left(\mathbf{C}^{e,i}_A + \mathbf{C}^{e,i}_B - \mathbf{C}^{e,i}_{AB} - \mathbf{C}^{e,i}_{BA}\right)^{-1} .$$

This fusion rule has already been applied at the beginning of this chapter in order to derive and study the joint density (25) for independent local Gaussian mixture estimates. In the following example, this fusion rule is applied to the joint Gaussian mixtures of the previous examples.

Example: Fusion Results After Global and Local Processing
Figures 1(b), 2(b), and 3(c) have been computed by employing the derived fusion rule for Gaussian mixture representations. In particular, Fig. 3(c) elucidates that local processing and subsequent fusion differs from a global processing scheme and only provides suboptimal results.

Discussion: A management system for distributed track-to-track fusion applications must be in the position to reconstruct the joint Gaussian mixture density from the local tracks. While linear Gaussian estimation problems only require knowledge on the cross-covariance matrices, the step towards Gaussian mixtures renders distributed estimation much more difficult: The cross-covariance matrices for every component have to be tracked and for this, a one-to-one assignment between the local components is required. For instance, the exemplary process noise (30) has indicated that different covariance parameters between different components of $f(\underline{x}_A)$ and $f(\underline{x}_B)$ can appear. Furthermore, the weighting factors for the joint Gaussian mixture have to be reconstructed, which are dependent on the actual measurements. A solution could consist of storing local and global weights separately. The local weights are updated by local measurements, which do not require information about other estimates. The global weights are used for the prediction step in order to prevent double counting of the weights of the transition density.

VI. Conservative Fusion Strategies

The preceding section has unveiled that bookkeeping of the nonlinear dependency structure between Gaussian mixture densities is far more difficult than between Gaussian densities. If bookkeeping is too expensive or fully decentralized scenarios are considered, suboptimal fusion strategies can be pursued that do not require knowledge about the underlying dependencies. The exponential mixture fusion rule [27]–[29]

$$f_{\text{EMD}}(\underline{x}) = C\, f_A^{\alpha}(\underline{x})\, f_B^{1-\alpha}(\underline{x}) \ , \quad \alpha \in (0,1)$$

can be viewed as the most notable concept, which is a generalization of covariance intersection [10] or covariance inflation [30]. This fusion rule provides conservative results according to [29]. By raising the Gaussian mixtures (23) and (24) to the power of α and $(1-\alpha)$, the fusion result is not a Gaussian mixture anymore. In [12], [31], it has been proposed to apply covariance intersection to each components of the Gaussian mixtures to be fused, which gives

$$f_{\text{GCI}}(\underline{x}) = \sum_{i=1}^{N_A} \sum_{i=1}^{N_B} \omega_A^i \omega_B^j \mathcal{N}\left(\begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_A^i \\ \hat{\underline{x}}_B^j \end{bmatrix}, \begin{bmatrix} \frac{1}{\alpha}\mathbf{C}_A^i & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\alpha}\mathbf{C}_B^j \end{bmatrix} \right) ,$$

where the covariance matrices are upper bounds for the covariance matrices in (34) (see, e.g. [30]). Both strategies may result into very conservative fusion results.

A different possibility is to conservatively bound the common process noise, such that no dependencies between the local estimates arise. This strategy of federated filtering [13] can be

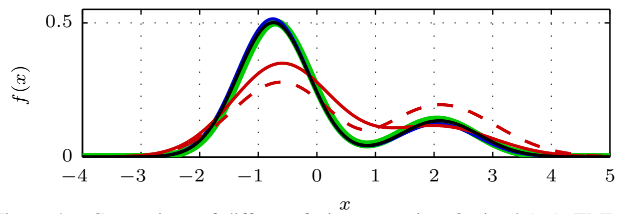


Figure 4. Comparison of different fusion strategies: Optimal (—), EMD (- -), GCI (—), NF (—), FGM (—).

applied as long as no other sources of dependencies are present. Instead of the joint transition density $f(\underline{y}_A, \underline{y}_B \,|\, \underline{x}_A, \underline{x}_B)$ in (28), the conservative product representation

$$f_{\text{NF}}(\underline{y}_A, \underline{y}_B \,|\, \underline{x}_A, \underline{x}_B) = f^{\alpha}(\underline{y}_A \,|\, \underline{x}_A,) \cdot f^{1-\alpha}(\underline{y}_B \,|\, \underline{x}_B)$$

is used such that each local density can independently be processed in the prediction step. However, this concept again does not preserve the Gaussian mixture representation. Sec. V has demonstrated that the process noise density for the joint Gaussian mixture consists of fully correlated components. Hence, a different conservative approximation

$$f_{\text{FGM}}(\underline{w}) = \sum_{i=1}^{N^w} \tilde{w}^{w,i} \mathcal{N}\left(\begin{bmatrix} \underline{w} \\ \underline{w} \end{bmatrix}; \begin{bmatrix} \hat{\underline{w}}^i \\ \hat{\underline{w}}^i \end{bmatrix}, \begin{bmatrix} \frac{1}{\alpha^i}\mathbf{C}^{w,i} & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\alpha^i}\mathbf{C}^{w,i} \end{bmatrix} \right)$$

of the common process noise (27) can be obtained by conservatively bounding each component. This way of treating the common process noise preserves the Gaussian mixture representation of the joint estimate and fusion results. However, it requires bookkeeping of the weighting coefficients. A concluding example provides a comparison of the different fusion strategies.

Example: Suboptimal Fusion Strategies
Five further prediction and local filtering steps are performed based on the result in Fig. 3(b). In each prediction step, the process noise (30) is used. At sensor node A, the local likelihood (33) is applied. Sensor node B employs the likelihood

$$f(z_B \,|\, x_B) = 0.1\,\mathcal{N}\big(x_B, 0, \tfrac{1}{2}\big) + 0.9\,\mathcal{N}\big(x_B, 3, \tfrac{2}{5}\big) .$$

Fig. 4 depicts different fusion strategies. The green density is the optimal fusion result that is based on a joint Gaussian mixture with 1024 components. Each local Gaussian mixture has only 128 components. Note that the joint Gaussian mixture has less than 128^2 component due to the fully dependent process noise. The red dashed density is the exponential mixture fusion result that is very conservative and not a Gaussian mixture. Componentwise application of covariance intersection yields the red density, which is also conservative but a Gaussian mixture. The blue density is obtained by the nonlinear federated filter, which only bounds the process noise and yields close-to-optimal results. The black density employs the conservative process noise approximation by a Gaussian mixture and hence is itself a Gaussian mixture. α is 0.5 in each case.

VII. Conclusions

This paper has demonstrated how to construct and exploit the dependency structure for Gaussian mixture estimates. The considerations prove that nonlinear track-to-track fusion is far more difficult than in the linear case. Not only the cross-covariance matrices between every two components have to be stored and updated in each processing step, but also the weighting parameters have to be kept track of. In particular, the weights are the reason why efficient distributed estimation algorithms are difficult to achieve. The weights cannot be managed independently from the local track estimates since they depend on the actual measurements. In addition, the weights of the process noise must not be double-counted. An idea is to store weights that are related to local measurements separately from weights that are related to the process noise. A further problem is the continuously increasing number of components, which has to be addressed by reduction algorithms. In this regard, it has to be analyzed how these algorithms affect the dependency structure.

So far, only the two-sensor case has been studied. For more than two sensor nodes, the dependencies between every two Gaussian mixtures have to be considered, which may lead to unacceptably large demands for memory and computational power. In general, this paper encourages the assumption that other density representations for nonlinear estimation also do not possess simpler dependency structures.

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] 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.

[4] 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.

[5] F. Govaers and W. Koch, "Distributed Kalman Filter Fusion at Arbitrary Instants of Time," in *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, UK, Jul. 2010.

[6] 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.

[7] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of Basic Engineering*, pp. 35–45, 1960.

[8] D. Willner, C. B. Chang, and K. P. Dunn, "Kalman Filter Algorithms for a Multi-sensor System," in *Proceedings of the 15th IEEE Conference on Decision and Control (CDC 1976)*, Clearwater, Florida, USA, Dec. 1976.

[9] 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.

[10] 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.

[11] 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.

[12] B. Upcroft, L.-L. Ong, S. Kumar, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Rich Probabilistic Representations for Bearing Only Decentralised Data Fusion," in *Proceedings of the 8th International Conference on Information Fusion (Fusion 2005)*, Philadelphia, Pennsylvania, USA, Jul. 2005.

[13] B. Noack, S. J. Julier, M. Reinhardt, and U. D. Hanebeck, "Nonlinear Federated Filtering," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[14] 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.

[15] C.-Y. Chong and S. Mori, "Graphical Models for Nonlinear Distributed Estimation," in *Proceedings of the 7th International Conference on Information Fusion (Fusion 2004)*, Stockholm, Sweden, Jun. 2004.

[16] 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.

[17] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[18] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active Learning with Statistical Models," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 129–145, 1996.

[19] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[20] J. Ajgl and M. Šimandl, "On Conservativeness of Posterior Density Fusion," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

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

[22] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[23] X.-R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[24] L.-L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Consistent Methods for Decentralised Data Fusion using Particle Filters," in *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Heidelberg, Germany, Sep. 2006, pp. 85–91.

[25] F. Sawo, D. Brunn, and U. D. Hanebeck, "Parameterized Joint Densities with Gaussian Mixture Marginals and their Potential Use in Nonlinear Robust Estimation," in *Proceedings of the 2006 IEEE International Conference on Control Applications (CCA 2006)*, Munich, Germany, Oct. 2006.

[26] F. Sawo, D. Brunn, and U. D. Hanebeck, "Parameterized Joint Densities with Gaussian and Gaussian Mixture Marginals," in *Proceedings of the 9th International Conference on Information Fusion (Fusion 2006)*, Florence, Italy, Jul. 2006.

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

[28] 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, UK, Sep. 2006, pp. 160–163.

[29] 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.

[30] 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.

[31] S. J. Julier, "An Empirical Study into the Use of Chernoff Information for Robust, Distributed Fusion of Gaussian Mixture Models," in *Proceedings of the 9th International Conference on Information Fusion (Fusion 2006)*, Florence, Italy, Jul. 2006.