Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion

Benjamin Noack, Joris Sijs, Marc Reinhardt, Uwe D. Hanebeck

CONTENTS

1.1 Introduction

Distributed and decentralized processing and fusion of sensor data is increasingly gaining in importance. In view of the Internet of Things and the vision of ubiquitous sensing, designing and implementing multisensor state estimation algorithms has already become a key issue. A network of interconnected sensor devices is usually characterized by the idea to process and collect data locally and independently on the sensor nodes. However, this does not imply that the data are independent of each other, and the state estimation algorithms have to address possible interdependencies so as to avoid erroneous data fusion results.

Dependencies among local estimates generally can be traced back to common sensor information and common process noise. A wide variety of Kalman filtering schemes allow for the treatment of dependent data in centralized, distributed, and decentralized networks of sensor nodes, but making the right choice is itself dependent upon analyzing and weighing up the different advantages and disadvantages. This chapter discusses different strategies to identify and treat dependencies among Kalman filter estimates while pointing out advantages and challenges.

The study commences with an analysis of the sources of dependencies between state estimates and with the optimal estimation strategy, which consists of a centralized processing of sensor data. Although an efficient preprocessing of sensor data is possible, the central node must have access to each sensor measurement at each processing step. Therefore, centralized schemes may lead to unacceptably high rates and volumes of data transfers. Distributing storage and computation over the network often proves to be an alternative. With the concept of federated Kalman filtering, dependencies stemming from a common process noise model can be treated; also an optimal distributed Kalman filter can be achieved when strict prerequisites are ensured to be met. These requirements can successively be relaxed by employing hypotheses or by striving for a consensus among sensor nodes. In a fully decentralized network with nodes operating autonomously, an optimal fusion strategy is attainable if the underlying dependencies are accessible. However, bookkeeping of dependencies is often unacceptably expensive in terms of both performance and memory usage. In this regard, the focus lies on suboptimal estimation and fusion strategies that do not require precise information about the underlying dependency structure but employ conservative bounds on the dependencies. The concept of ellipsoidal intersection can be employed to counteract the problem of double-counting sensor data. Fusion of estimates by means of covariance intersection is most conservative but also most insusceptible to unmodeled dependencies and does not require any information about the underlying dependencies. Estimation quality and performance of these estimation concepts are considered, and it is discussed when and how to use them.

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 $(\hat{\underline{x}}, \mathbf{C})$, we denote an estimate with mean $\hat{\underline{x}}$ and covariance matrix $\mathbf{C}$. The notation $\hat{\underline{x}}$ is used for the mean of a random variable, an estimate of an uncertain quantity, or an observation. The matrix $\mathbf{I}$ denotes the identity matrix of appropriate dimension. 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.

1.2 State Estimation in Networked Systems

State estimation methods are utilized to provide insights into a system's behavior. An estimate for the system's state is dynamically computed based on prior information, a process model, and measurements stemming from sensor devices. The system is characterized by a discrete-time linear process model

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}_k \, \underline{\boldsymbol{x}}_k + \mathbf{B}_k \, \hat{\underline{u}}_k + \underline{\boldsymbol{w}}_k \ , \tag{1.1}$$

where the system matrix $\mathbf{A}_k \in \mathbb{R}^{n_x \times n_x}$ maps the state vector $\underline{\boldsymbol{x}}_k \in \mathbb{R}^{n_x}$ from time step k to the subsequent step $k+1$. The temporal evolution can further be affected by a control input $\hat{\underline{u}}_k \in \mathbb{R}^{n_u}$ with control-input matrix $\mathbf{B}_k \in \mathbb{R}^{n_x \times n_u}$ and a white Gaussian process noise $\underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^w)$ with covariance matrix $\mathbf{C}_k^w \in \mathbb{R}^{n_x \times n_x}$.

In order to derive an estimate for the system's state, measurement data from sensor devices have to be acquired and processed. A multisensor system typically collects a vast number of measurements $\hat{\underline{z}}_k^i \in \mathbb{R}^{n_z}$, $i \in \{1, \ldots, N\}$. An observation $\hat{\underline{z}}_k^i$ provided by sensor device i at time k is related to the state through the linear measurement model

$$\underline{\boldsymbol{z}}_k^i = \mathbf{H}_k^i \, \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k^i \ , \tag{1.2}$$

i.e., $\hat{\underline{z}}_k^i$ is a realization of $\underline{\boldsymbol{z}}_k^i$, where $\mathbf{H}_k^i \in \mathbb{R}^{n_z \times n_x}$ is the measurement matrix and $\underline{\boldsymbol{v}}_k^i \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^{z,i})$ denotes a zero-mean white sensor noise with error covariance matrix $\mathbf{C}_k^{z,i} \in \mathbb{R}^{n_z \times n_z}$.

1.2.1 Kalman Filtering

With the models (1.1) and (1.2) having a linear structure and being affected by Gaussian noise terms, state estimates for $\underline{\boldsymbol{x}}_k$ can be computed recursively and in closed form in terms of the Kalman filter formulas. The Kalman filter algorithm [14], moreover, embodies an optimal solution to the state estimation problem providing estimates that minimize the mean-squared estimation error. The Kalman filter calculates the parameters $(\hat{\underline{x}}_k^{\mathrm{e}}, \mathbf{C}_k^{\mathrm{e}})$ where $\hat{\underline{x}}_k^{\mathrm{e}} \in \mathbb{R}^{n_x}$ is the estimate at time k given all previous and current observations and $\mathbf{C}_k^{\mathrm{e}} \in \mathbb{R}^{n_x \times n_x}$ is the corresponding error covariance matrix

$$\mathbf{C}_k^{\mathrm{e}} = \mathrm{E}[(\hat{\underline{x}}_k^{\mathrm{e}} - \underline{\boldsymbol{x}}_k)(\hat{\underline{x}}_k^{\mathrm{e}} - \underline{\boldsymbol{x}}_k)^{\mathrm{T}}] \ .$$

Its trace yields the mean-squared estimation error. The recursive processing of measurement data is carried out by combining the measurement information with prior information at each time k. The Kalman filter scheme is therefore composed of prediction and filtering steps. In the prediction step, the process model (1.1) is employed to provide prior information for the subsequent time step $k+1$. The prediction result $(\hat{\underline{x}}^{\mathrm{p}}_{k+1}, \mathbf{C}^{\mathrm{p}}_{k+1})$ is computed according to

$$\hat{\underline{x}}^{\mathrm{p}}_{k+1} = \mathbf{A}_k \, \hat{\underline{x}}^{\mathrm{e}}_k + \mathbf{B}_k \, \hat{\underline{u}}_k \tag{1.3}$$

and

$$\mathbf{C}^{\mathrm{p}}_{k+1} = \mathbf{A}_k \mathbf{C}^{\mathrm{e}}_k \mathbf{A}^{\mathrm{T}}_k + \mathbf{C}^{w}_k \ . \tag{1.4}$$

The predicted estimate serves as prior information in the filtering step where it is combined with a current measurement vector $\hat{\underline{z}}^i_k$, which is related to the state by model (1.2). The update formulas are

$$\begin{aligned} \hat{\underline{x}}^{\mathrm{e}}_k &= \hat{\underline{x}}^{\mathrm{p}}_k + \mathbf{K}^i_k (\hat{\underline{z}}^i_k - \mathbf{H}^i_k \hat{\underline{x}}^{\mathrm{p}}_k) \\ &= (\mathbf{I} - \mathbf{K}^i_k \mathbf{H}^i_k) \hat{\underline{x}}^{\mathrm{p}}_k + \mathbf{K}^i_k \, \hat{\underline{z}}^i_k \end{aligned} \tag{1.5}$$

for the state estimate and

$$\begin{aligned} \mathbf{C}^{\mathrm{e}}_k &= (\mathbf{I} - \mathbf{K}^i_k \mathbf{H}^i_k) \mathbf{C}^{\mathrm{p}}_k (\mathbf{I} - \mathbf{K}^i_k \mathbf{H}^i_k)^{\mathrm{T}} + \mathbf{K}^i_k \mathbf{C}^{z,i}_k (\mathbf{K}^i_k)^{\mathrm{T}} \\ &= \mathbf{C}^{\mathrm{p}}_k - \mathbf{K}^i_k \mathbf{H}^i_k \mathbf{C}^{\mathrm{p}}_k \ . \end{aligned} \tag{1.6}$$

for the corresponding error covariance matrix. In both formulas, the Kalman gain

$$\begin{aligned} \mathbf{K}^i_k &= \mathbf{C}^{\mathrm{p}}_k (\mathbf{H}^i_k)^{\mathrm{T}} \big(\mathbf{C}^{z,i}_k + \mathbf{H}^i_k \mathbf{C}^{\mathrm{p}}_k (\mathbf{H}^i_k)^{\mathrm{T}} \big)^{-1} && (1.7) \\ &= \big((\mathbf{C}^{\mathrm{p}}_k)^{-1} + (\mathbf{H}^i_k)^{\mathrm{T}} (\mathbf{C}^{z,i}_k)^{-1} \mathbf{H}^i_k \big)^{-1} (\mathbf{H}^i_k)^{\mathrm{T}} (\mathbf{C}^{z,i}_k)^{-1} && (1.8) \end{aligned}$$

is employed, which ensures an optimal combination in terms of a minimum mean-squared error. These formulas are apparently not restricted to a single measurement per time step; equations (1.5) to (1.7) can simply be repeated for each available measurement. However, major difficulties arise when multiple sensors provide measurements over a network or multiple state estimation systems provide estimates to be fused.

1.2.2 Multisensor Systems and Networks

In line with the rapid advances made in sensor and network technologies, there is a rising demand for distributed implementations of Kalman filter algorithms. The intended benefits include scalability, mobility, and robustness to failures. Sometimes, it is simply the need to use networked systems, e.g., in order to monitor a large-scale phenomenon, that calls for distributed state estimation algorithms. From a visionary perspective, a network of sensor systems is capable of managing vast amounts of data, automatically responding to unpredictable changes of network conditions and environment, and reorganizing itself. In general, the design of state estimation algorithms cannot reflect these expectations without endangering reliability and compromising quality of estimates.

Apparently, distributed state estimation algorithms have to take into account the network architecture and are subject to network effects such as packet losses and delays. However, even a fully reliable network poses challenges to the task of estimating the system's state. Section 1.3 reveals dependencies between locally processed data as a major challenge for networked state estimation. Solutions to this challenge have to be oriented towards the underlying network architecture. More precisely, the state estimation algorithm depends on where and when measurement data is to be processed. On the basis of different, basic estimation architecture, sections 1.4, 1.5, and 1.6 discuss several approaches to overcome these challenges.

1.3 Sources of Dependent Information

The standard formulation of the Kalman filter in Sec. 1.2.1 assumes conditional independence of measurements given the state, white process and measurement noise, and just-in-time processing. Many practical applications prevent these assumptions from being satisfied. In the last decades, much effort has been spent on making the Kalman filter robust to realistic violations of these assumptions. Extensions towards the treatment of dependencies due to colored noise terms, for instance, have been addressed in [31] by state augmentations. A processing of out-of-sequence measurements that may arise from packet delays in a network can be achieved by accumulated state densities [16] or delayed-state representations [10]. In general, the violated assumptions lead to correlations, i.e., dependencies, that have to be properly treated. For example, an out-of-sequence measurement cannot be deemed to be conditionally independent of the current state anymore.

If dependencies across noise terms are known, the Kalman filter algorithm can be altered so as to still provide an optimal estimate. However, dependencies across different pieces of information in a networked system are often difficult to keep track of and to trace back. In particular, identifying dependencies becomes a serious problem if the network not only collects sensor data, but also the Kalman filter algorithm itself is distributed over the network [20]. While distributed and decentralized Kalman filter implementations offer the advantage of handing over workload to the sensor nodes and of reducing the overall communication load, strong correlations among the outputs of the nodes can occur. The reason for this lies in the local acquisition and (pre-)processing of sensor data that is required to distribute the processing steps of the Kalman filter. In essence, the path data take to traverse the network and the redundant use of models are sources of underlying correlations.

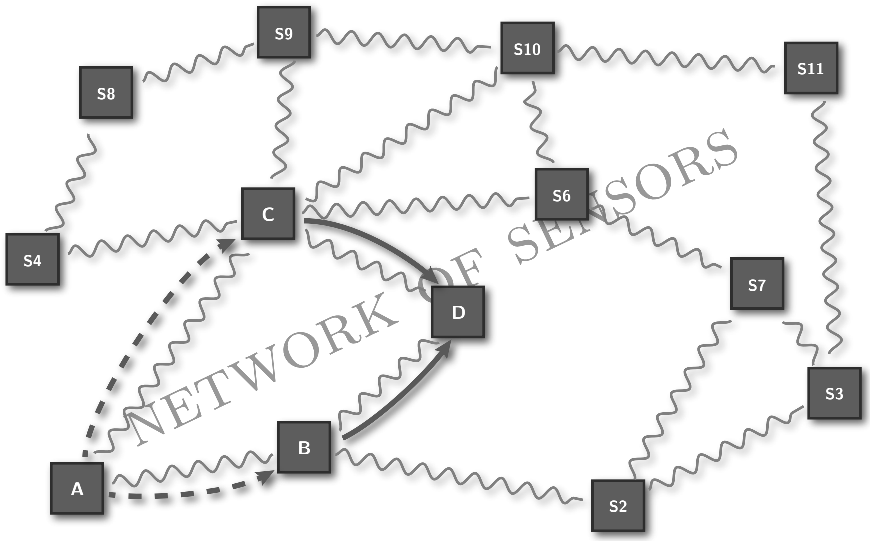

FIGURE 1.1
Sensor node D is not aware of data from node A that has been processed in node B as well as in node C.

1.3.1 Double-counting of Common Sensor Data

Double counting of data can be viewed as the most apparent reason for dependencies between data sets of different sensor nodes [7]. In order to establish efficient and communication-saving data transfers, the nodes do not simply transmit raw sensor data but usually preprocess and fuse received data before forwarding them. This leads to the problem that already processed information is concealed in the processing results and might erroneously be processed multiple times. Such a situation is depicted in Fig. 1.1 where node A sends its data to node B and node C. Both node B and node C further process the received data and transmit their results to node D. Node D is not in the position to sort out that data from node A are included in both received data sets. Processing in node D possibly leads to double-counting of data from node A. As shown in the following example, double-counting leads to correlations.

> **Example:** Suppose that each node observes the state $\underline{\boldsymbol{x}}$ with identity measurement matrix $\mathbf{H} = \mathbf{I}$, where each measurement serves as an estimate, i.e., $\hat{\underline{x}} = \hat{\underline{z}}$. Sensor node A provides a measurement $\hat{\underline{x}}_{\mathsf{A}}$ with error covariance matrix $\mathbf{C}_{\mathsf{A}}$. This information is transmitted to nodes B and C, which themselves have the observations $\hat{\underline{x}}_{\mathsf{B}}$ and $\hat{\underline{x}}_{\mathsf{C}}$, respectively. For the purpose of reducing the amount of data to be transferred, both nodes combine the received measurement with

their own measurement, i.e., $\tilde{\underline{x}}_\mathsf{B} = (\mathbf{I} - \mathbf{K}_\mathsf{B})\,\hat{\underline{x}}_\mathsf{B} + \mathbf{K}_\mathsf{B}\,\hat{\underline{x}}_\mathsf{A}$ and $\tilde{\underline{x}}_\mathsf{C} = (\mathbf{I} - \mathbf{K}_\mathsf{C})\,\hat{\underline{x}}_\mathsf{C} + \mathbf{K}_\mathsf{C}\,\hat{\underline{x}}_\mathsf{A}$, where $\mathbf{K}_\mathsf{B}$ and $\mathbf{K}_\mathsf{C}$ are some locally determined gains. A node D that receives $\tilde{\underline{x}}_\mathsf{B}$ and $\tilde{\underline{x}}_\mathsf{C}$ without further information is not aware of the correlations between them due to $\hat{\underline{x}}_\mathsf{A}$. They are given by the cross-covariance term $\mathrm{E}[(\tilde{\underline{x}}_\mathsf{B} - \underline{\boldsymbol{x}})(\tilde{\underline{x}}_\mathsf{C} - \underline{\boldsymbol{x}})^\mathrm{T}] = \mathbf{K}_\mathsf{B}\mathbf{C}_\mathsf{A}\mathbf{K}_\mathsf{C}^\mathrm{T}$.

1.3.2 Common Prior Information and Process Noise

The second important source of dependencies is caused by parallel processing and modeling of the same noise terms. In particular, several Kalman filters that run in parallel on the same estimation problem automatically output correlated estimates. Correlations may already arise at the early initialization stage of the local Kalman filters, i.e., an initialization of each estimator with the same prior estimate implies a full correlation between the local estimates. Besides possible common prior information, each Kalman filter usually employs the same state transition model (1.1) and hence incorporates the same process noise in each prediction step. The problem of common process noise is especially encountered in track-to-track fusion applications [1, 2] and decentralized state estimation [13]. The effect of common process noise is elucidated in the following example.

Example: At time k, two local Kalman filter implementations provide the estimates $\hat{\underline{x}}_\mathsf{A}^\mathrm{e}$ and $\hat{\underline{x}}_\mathsf{B}^\mathrm{e}$ on $\underline{\boldsymbol{x}}_k$ that are supposed to have independent estimation errors, i.e., $\mathbf{C}_\mathsf{AB}^\mathrm{e} = \mathrm{E}[(\hat{\underline{x}}_\mathsf{A}^\mathrm{e} - \underline{\boldsymbol{x}}_k)(\hat{\underline{x}}_\mathsf{B}^\mathrm{e} - \underline{\boldsymbol{x}}_k)^\mathrm{T}] = \mathbf{0}$. The local prediction steps, according to (1.3), yield $\hat{\underline{x}}_\mathsf{A}^\mathrm{p} = \mathbf{A}_k\,\hat{\underline{x}}_\mathsf{A}^\mathrm{e}$ and $\hat{\underline{x}}_\mathsf{B}^\mathrm{p} = \mathbf{A}_k\,\hat{\underline{x}}_\mathsf{B}^\mathrm{e}$ (no input $\hat{\underline{u}}_k$). The predicted estimates now have correlated errors, and the cross-covariance matrix yields

$$\begin{aligned}\mathbf{C}_\mathsf{AB}^\mathrm{p} &= \mathrm{E}[(\hat{\underline{x}}_\mathsf{A}^\mathrm{p} - \underline{\boldsymbol{x}}_k)(\hat{\underline{x}}_\mathsf{B}^\mathrm{p} - \underline{\boldsymbol{x}}_k)^\mathrm{T}] \\ &= \mathrm{E}\left[\left(\mathbf{A}_k\,\hat{\underline{x}}_\mathsf{A}^\mathrm{e} - (\mathbf{A}_k\,\underline{\boldsymbol{x}}_k + \underline{\boldsymbol{w}}_k)\right)\left(\mathbf{A}_k\,\hat{\underline{x}}_\mathsf{B}^\mathrm{e} - \left(\mathbf{A}_k\,\underline{\boldsymbol{x}}_k + \underline{\boldsymbol{w}}_k\right)\right)^\mathrm{T}\right] \\ &= \mathbf{A}_k\,\mathbf{C}_\mathsf{AB}^\mathrm{e}\,\mathbf{A}_k^\mathrm{T} + \mathbf{C}_k^w = \mathbf{C}_k^w\ .\end{aligned}$$

The prediction step hence causes dependencies, which appear in form of the process noise covariance matrix.

1.3.3 Strategies for Networked State Estimation

In order to appropriately treat dependencies arising from in-network processing of estimates, the design of the Kalman filter algorithm heavily depends on and has to be adapted to the underlying and desired network architecture. Often, a coarse classification into centralized, distributed, and decentralized estimation architectures is proposed, which takes into consideration where sensor data is processed, when data is to be transmitted, and which knowledge nodes share. With this classification, different basic multisensor Kalman filter concepts can accordingly be categorized [17, 20].

In a centralized architecture, a single node is charged with the task of processing all the sensor data recorded in the network and computing an estimate. Hence, all measurements have to be transmitted to a central node, which feeds a Kalman filter with the entire data. Sec. 1.4 discusses multisensor filtering principles. In order to reduce the amount of data to be communicated and to relieve workload of the center node, distributed implementations of the Kalman filter can be pursued. For distributed Kalman filtering, cooperative nodes are typically required, and often still a central processing unit is present to assemble a global estimate from the local processing results. Sec. 1.5 points out that distributed estimation involves a tradeoff between an optimal estimation quality and a high robustness to node and communication failures. A decentralized architecture differs from a distributed one in that the estimation problem is solved locally on each sensor node. The nodes are usually designed to operate independently of each other and can share their information in order to solve the estimation problem on a global level. While the nodes optimize the estimation quality on a local scale, a decentralized network generally cannot reach the estimation quality achieved by centralized or distributed architectures, because dependencies between the local estimates often cannot be reconstructed and exploited. Decentralized Kalman filtering lies in the focus of Sec. 1.6.

1.4 Centralized Multisensor State Estimation

A centralized processing of the entire sensor data offers the advantage that the conditional independence of the measurements can still be exploited and, in particular, the standard Kalman filter formulas can be used. Therefore, the problem of common process noise cannot occur in this setup, and double-counting of data is avoided, for example, by labeling data. By contrast, each node must transmit its sensor data to the central processing unit, which can lead to high traffic, particularly in multi-hop networks. The first part of this section discusses the processing of multisensor data within the Kalman filter. In the second part, the information form is used to efficiently preprocess sensor data during transmission in order to reduce the amount of data to be transmitted.

1.4.1 Multisensor Kalman Filtering

For the processing of multiple sensor data, two possibilities can essentially be named. The set $\mathcal{Z}_k = \{\hat{\underline{z}}_k^1, \ldots, \hat{\underline{z}}_k^N\}$ of measurements that are received at a time step k can be processed either sequentially or en-bloc [3]. Each measurement is related to the state through the corresponding model (1.2). For a sequential processing as illustrated in Fig. 1.2, the Kalman filter formulas (1.5), (1.6), and (1.7) are applied recursively to each measurement, i.e.,

$$\begin{aligned}
\underline{\tilde{x}}_k^{\mathrm{e},1} &= (\mathbf{I} - \mathbf{K}_k^1 \mathbf{H}_k^1)\underline{\hat{x}}_k^{\mathrm{p}} + \mathbf{K}_k^1\, \underline{\hat{z}}_k^1 , & \widetilde{\mathbf{C}}_k^{\mathrm{e},1} &= \mathbf{C}_k^{\mathrm{p}} - \mathbf{K}_k^1 \mathbf{H}_k^1 \mathbf{C}_k^{\mathrm{p}} , \\
\underline{\tilde{x}}_k^{\mathrm{e},2} &= (\mathbf{I} - \mathbf{K}_k^2 \mathbf{H}_k^2)\underline{\tilde{x}}_k^{\mathrm{e},1} + \mathbf{K}_k^2\, \underline{\hat{z}}_k^2 , & \widetilde{\mathbf{C}}_k^{\mathrm{e},2} &= \widetilde{\mathbf{C}}_k^{\mathrm{e},1} - \mathbf{K}_k^2 \mathbf{H}_k^2\, \widetilde{\mathbf{C}}_k^{\mathrm{e},1} , \\
&\vdots & &\vdots \\
\underline{\hat{x}}_k^{\mathrm{e}} &= (\mathbf{I} - \mathbf{K}_k^N \mathbf{H}_k^N)\underline{\tilde{x}}_k^{\mathrm{e},N-1} + \mathbf{K}_k^N\, \underline{\hat{z}}_k^N , & \mathbf{C}_k^{\mathrm{e}} &= \widetilde{\mathbf{C}}_k^{\mathrm{e},N-1} - \mathbf{K}_k^N \mathbf{H}_k^N\, \widetilde{\mathbf{C}}_k^{\mathrm{e},N-1}
\end{aligned}$$

with $\mathbf{K}_k^i = \widetilde{\mathbf{C}}_k^{\mathrm{e},i-1} (\mathbf{H}_k^i)^{\mathrm{T}} \big(\mathbf{C}_k^{z,i} + \mathbf{H}_k^i \widetilde{\mathbf{C}}_k^{\mathrm{e},i-1} (\mathbf{H}_k^i)^{\mathrm{T}}\big)^{-1}$. The last equation yields the filtering result, which is independent of the order of processing the measurements.

Instead of performing N filtering steps, the measurement update can also be carried out in a single step. For this, the measurements are concatenated into a single vector $\underline{\hat{z}}_k^{\mathrm{total}} := [(\underline{\hat{z}}_k^1)^{\mathrm{T}}, \ldots, (\underline{\hat{z}}_k^N)^{\mathrm{T}}]^{\mathrm{T}}$. Similarly, the matrices $\mathbf{H}_k^{\mathrm{total}}$ and $\mathbf{C}_k^{z,\mathrm{total}}$ have to be compiled from local measurement and covariance matrices in order to compute the en-bloc filtering result

$$\begin{aligned}
\underline{\hat{x}}_k^{\mathrm{e}} &= (\mathbf{I} - \mathbf{K}_k^{\mathrm{total}} \mathbf{H}_k^{\mathrm{total}})\, \underline{\hat{x}}_k^{\mathrm{p}} + \mathbf{K}_k^{\mathrm{total}}\, \underline{\hat{z}}_k^{\mathrm{total}} , \\
\mathbf{C}_k^{\mathrm{e}} &= \mathbf{C}_k^{\mathrm{p}} - \mathbf{K}_k^{\mathrm{total}} \mathbf{H}_k^{\mathrm{total}} \mathbf{C}_k^{\mathrm{p}} ,
\end{aligned}$$

where the gain is given by

$$\mathbf{K}_k^{\mathrm{total}} = \mathbf{C}_k^{\mathrm{p}} \big(\mathbf{H}_k^{\mathrm{total}}\big)^{\mathrm{T}} \big(\mathbf{C}_k^{z,\mathrm{total}} + \mathbf{H}_k^{\mathrm{total}} \mathbf{C}_k^{\mathrm{p}} (\mathbf{H}_k^{\mathrm{total}})^{\mathrm{T}}\big)^{-1} .$$

This scheme is depicted in Fig. 1.3.

Both processing schemes require that all measurements taken at time step k are available at the center node. With the help of state augmentations [16], also delayed measurements can be processed and also less frequent data transfers can be established. However, the amount of data to be transmitted is not reduced. A further problem in multi-hop networks is that nodes without a direct link to the center node have to pass on their data to other nodes. Hence, nodes in proximity to the center node might suffer from a high load of transfers. The information form presented in the following section allows to efficiently condense data along any communication path.

1.4.2 Information Filtering

The processing of multiple measurements consists of either nested Kalman filtering steps or constructing a joint measurement vector and mapping. Both ways are cumbersome for the central node. A distributable version of the filtering step can be achieved by the information form [18] of the Kalman filter, which became very popular in multisensor estimation problems and

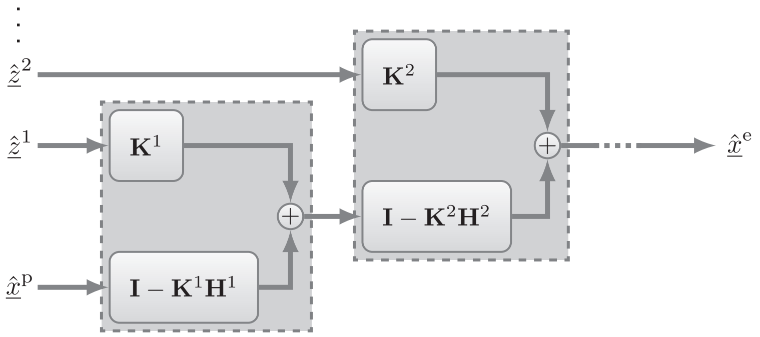
$\underline{\hat{z}}^2$
$\underline{\hat{z}}^1$
$\underline{\hat{x}}^{\mathrm{p}}$
$\mathbf{K}^1$
$\mathbf{I} - \mathbf{K}^1\mathbf{H}^1$
$\mathbf{K}^2$
$\mathbf{I} - \mathbf{K}^2\mathbf{H}^2$
$+$
$\underline{\hat{x}}^{\mathrm{e}}$

FIGURE 1.2
Sequential multisensor Kalman filtering.

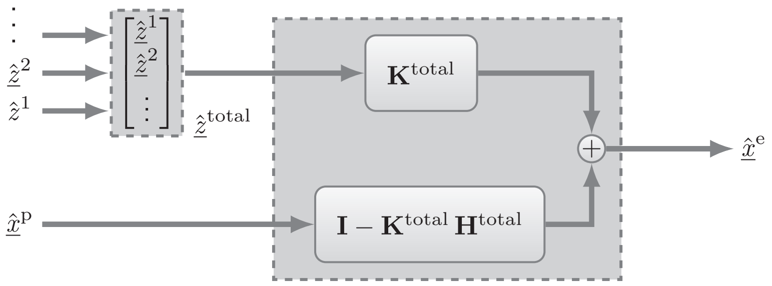
$\underline{\hat{z}}^2$
$\underline{\hat{z}}^1$
$\begin{bmatrix} \underline{\hat{z}}^1 \\ \underline{\hat{z}}^2 \\ \vdots \end{bmatrix}$
$\underline{\hat{z}}^{\mathrm{total}}$
$\mathbf{K}^{\mathrm{total}}$
$\mathbf{I} - \mathbf{K}^{\mathrm{total}}\,\mathbf{H}^{\mathrm{total}}$
$\underline{\hat{x}}^{\mathrm{p}}$
$\underline{\hat{x}}^{\mathrm{e}}$

FIGURE 1.3
Blockwise multisensor Kalman filtering.

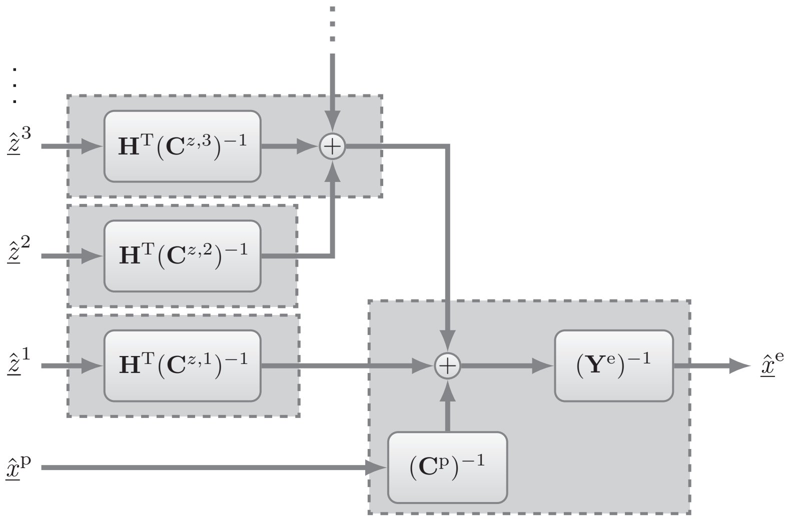
$\underline{\hat{z}}^3$
$\underline{\hat{z}}^2$
$\underline{\hat{z}}^1$
$\underline{\hat{x}}^{\mathrm{p}}$
$\mathbf{H}^{\mathrm{T}}(\mathbf{C}^{z,3})^{-1}$
$\mathbf{H}^{\mathrm{T}}(\mathbf{C}^{z,2})^{-1}$
$\mathbf{H}^{\mathrm{T}}(\mathbf{C}^{z,1})^{-1}$
$(\mathbf{C}^{\mathrm{p}})^{-1}$
$(\mathbf{Y}^{\mathrm{e}})^{-1}$
$\underline{\hat{x}}^{\mathrm{e}}$

FIGURE 1.4
Information filtering with preprocessing of sensor data.

essentially encompasses an algebraic reformulation. In place of the mean and covariance matrix, the information vector

$$\underline{\hat{y}}_k := \mathbf{C}_k^{-1} \underline{\hat{x}}_k$$

and the information matrix

$$\mathbf{Y}_k := \mathbf{C}_k^{-1}$$

are considered. The measurements are also transformed according to

$$\underline{i}_k^i = (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \underline{\hat{z}}_k^i \quad \text{and} \quad \mathbf{I}_k^i = (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \mathbf{H}_k^i \ .$$

The filtering step then only requires the computation of the simple sums

$$\underline{\hat{y}}_k^{\mathrm{e}} = \underline{\hat{y}}_k^{\mathrm{p}} + \sum_{i=1}^{N} \underline{i}_k^i = (\mathbf{C}_k^{\mathrm{p}})^{-1} \underline{\hat{x}}_k^{\mathrm{p}} + \sum_{i=1}^{N} (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \underline{\hat{z}}_k^i \tag{1.9}$$

and

$$\mathbf{Y}_k^{\mathrm{e}} = \mathbf{Y}_k^{\mathrm{p}} + \sum_{i=1}^{N} \mathbf{I}_k^i = (\mathbf{C}_k^{\mathrm{p}})^{-1} + \sum_{i=1}^{N} (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \mathbf{H}_k^i \tag{1.10}$$

for the information vector and information matrix, respectively. This reformulation offers the significant advantage that parts of the sum can be carried out along each communication path, i.e., each node can collect received information vectors, combine them with its own information vector, and just needs to forward a single information vector to the next node on the path.

The prediction step can also be formulated in terms of the information parameters $\underline{\hat{y}}_k^{\mathrm{e}}$ and $\mathbf{Y}_k^{\mathrm{e}}$. With $\mathbf{L}_{k+1} := \mathbf{A}_k (\mathbf{Y}_k^{\mathrm{e}})^{-1}$, the predicted information matrix becomes

$$\mathbf{Y}_{k+1}^{\mathrm{p}} = \left(\mathbf{L}_{k+1} \mathbf{Y}_k^{\mathrm{e}} \mathbf{L}_{k+1}^{\mathrm{T}} + \mathbf{B}_k \mathbf{C}_k^u \mathbf{B}_k^{\mathrm{T}} \right)^{-1} ,$$

and the predicted information vector yields

$$\underline{\hat{y}}_{k+1}^{\mathrm{p}} = \mathbf{Y}_{k+1}^{\mathrm{p}} \left(\mathbf{L}_{k+1} \, \underline{\hat{y}}_k^{\mathrm{e}} + \mathbf{B}_k \, \underline{\hat{u}}_k \right) .$$

The corresponding state estimate is obtained through the simple reverse transformations $\underline{\hat{x}}_k = \mathbf{Y}_k^{-1} \, \underline{\hat{y}}_k$ and $\mathbf{C}_k = \mathbf{Y}_k^{-1}$ for mean and covariance matrix, respectively.

1.5 Cooperative Distributed State Estimation

The previous section has pointed out that the filtering step already allows for a distributable reformulation by employing the inverse covariance form of the

TABLE 1.1
Overview of distributed Kalman filtering schemes.

method	local processing scheme	requirements for local processing	global estimate
federated Kalman filter	initialization and prediction with covariance inflation	upper bound on number of nodes	fusion at center node, suboptimal[1] (conservative)
distributed Kalman filter	initialization and prediction with covariance inflation, filtering with globalized covariance matrix	total number of nodes, all measurement and noise matrices	fusion at center node, optimal
hypothesizing Kalman filter	initialization and prediction with covariance inflation, filtering with globalization and correction matrix	hypothesis on global measurement capacity	fusion at center node, suboptimal[2]
consensus Kalman filter	filtering with additional consensus step on sensor data or estimates	data from neighboring nodes	synchronization, asymptotically optimal

[1]one-step optimal, [2]optimal if hypothesis is correct

Kalman filter. However, the central node must access the measurement data of each node and at each time step, which can turn out to be inefficient in terms of communication and computation load. Distributed implementations of the Kalman filter algorithm aspire to distribute the workload among the participating nodes. In general, a central node is still present that finally puts the nodes' data together into an estimate of the state. In contrast to the centralized multisensor Kalman filtering scheme, this is only necessary when an estimate is requested and thereby significantly reduces the communication rate. The main challenges that have to be addressed by the concepts in this section refer to dependencies that can be traced back to the initialization step and the problem of common process noise.

Distributed Kalman filtering, in general, requires a tradeoff between estimation quality and robustness to node and communication failures. Table 1.1 provides an overview of the concepts to be discussed in the following. In particular, it is highlighted which knowledge must be accessible by the nodes, which modifications to the local Kalman-filter-like processing schemes are necessary, and where a global estimate is computed.

1.5.1 Federated Kalman Filtering

The federated Kalman filter [4, 5] is grounded on the idea that each node runs its own Kalman filter, which computes an estimate based on the locally available sensor data. For a global estimate, the local Kalman filter results then have to be fused by the central node. As stated in Sec. 1.3.2, the local Kalman filters can neither be initialized with the same prior estimates nor employ the standard process model without causing strong correlations between their reported estimation errors. The federated Kalman filter therefore alters the local initialization and prediction steps such that the local estimates have independent error variances.

In order to avoid dependencies between the local processing results, the joint estimation error

$$\underline{\tilde{\boldsymbol{x}}}_k = \begin{bmatrix} \underline{\hat{x}}_k^{\mathrm{e},1} - \underline{\boldsymbol{x}}_k \\ \vdots \\ \underline{\hat{x}}_k^{\mathrm{e},N} - \underline{\boldsymbol{x}}_k \end{bmatrix} = \begin{bmatrix} \underline{\hat{x}}_k^{\mathrm{e},1} \\ \vdots \\ \underline{\hat{x}}_k^{\mathrm{e},N} \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} \underline{\boldsymbol{x}}_k$$

is considered. If the local estimator of each node $i \in \{1, \ldots, N\}$ is initialized with the same prior estimate $(\underline{\hat{x}}_0^{\mathrm{p},i}, \mathbf{C}_0^{\mathrm{p},i}) := (\underline{\hat{x}}_0^{\mathrm{p}}, \mathbf{C}_0^{\mathrm{p}})$, each cross-covariance matrix becomes $\mathrm{E}\left[(\underline{\hat{x}}_0^{\mathrm{p},i} - \underline{\boldsymbol{x}}_k)(\underline{\hat{x}}_0^{\mathrm{p},j} - \underline{\boldsymbol{x}}_k)^{\mathrm{T}}\right] = \mathbf{C}_0^{\mathrm{p}},\ i, j \in \{1, \ldots, N\}$. This implies that the joint error covariance matrix $\widetilde{\mathbf{C}}_0 = \mathrm{E}[\underline{\tilde{\boldsymbol{x}}}_0 \, \underline{\tilde{\boldsymbol{x}}}_0^{\mathrm{T}}]$ is fully occupied. The federated Kalman filter replaces this joint covariance matrix by the inflated joint covariance matrix

$$\begin{bmatrix} \frac{1}{\omega_1}\mathbf{C}_0^{\mathrm{p}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{1}{\omega_2}\mathbf{C}_0^{\mathrm{p}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \frac{1}{\omega_N}\mathbf{C}_0^{\mathrm{p}} \end{bmatrix} \geq \begin{bmatrix} \mathbf{C}_0^{\mathrm{p}} & \mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{C}_0^{\mathrm{p}} \\ \mathbf{C}_0^{\mathrm{p}} & \mathbf{C}_0^{\mathrm{p}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}_0^{\mathrm{p}} \\ \mathbf{C}_0^{\mathrm{p}} & \cdots & \mathbf{C}_0^{\mathrm{p}} & \mathbf{C}_0^{\mathrm{p}} \end{bmatrix} = \widetilde{\mathbf{C}}_0 \tag{1.11}$$

with $\sum_{i=1}^{N} \omega_i = 1$ and $\omega_i \geq 0$, which then implies that each sensor node is initialized with $(\underline{\hat{x}}_0^{\mathrm{p},i}, \mathbf{C}_0^{\mathrm{p},i}) := (\underline{\hat{x}}_0^{\mathrm{p}}, \frac{1}{\omega_i}\mathbf{C}_0^{\mathrm{p}})$. With the initial error covariance matrix $\mathbf{C}_0^{\mathrm{p}}$ multiplied by $\frac{1}{\omega_i}$, the initial estimates can then be deemed as independent. This technique is known as covariance inflation [8, 24] and is mainly used for the treatment of unknown correlations between estimation errors (see Sec. 1.6). The inflation parameters ω_i depend on the number N of participating nodes.

The federated Kalman filtering primarily aims at counteracting the problem of common process noise. As explained in Sec. 1.3.2, dependencies in form of the process noise covariance matrix $\mathbf{C}_k^w$ are caused by the prediction step, i.e., $\mathrm{E}\left[(\mathbf{A}_k \, \underline{\hat{x}}_k^{\mathrm{e},i} - \underline{\boldsymbol{x}}_{k+1})(\mathbf{A}_k \, \underline{\hat{x}}_k^{\mathrm{e},j} - \underline{\boldsymbol{x}}_{k+1})^{\mathrm{T}}\right] = \mathbf{C}_k^w,\ i \neq j$ is the cross-covariance matrix between two local, independent estimates $\underline{\hat{x}}_k^{\mathrm{e},i}$ and $\underline{\hat{x}}_k^{\mathrm{e},j}$ after prediction. For this reason, again an inflated version

$$\begin{bmatrix} \frac{1}{\omega_1}\mathbf{C}_k^w & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{1}{\omega_2}\mathbf{C}_k^w & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \frac{1}{\omega_N}\mathbf{C}_k^w \end{bmatrix} \geq \begin{bmatrix} \mathbf{C}_k^w & \mathbf{C}_k^w & \cdots & \mathbf{C}_k^w \\ \mathbf{C}_k^w & \mathbf{C}_k^w & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}_k^w \\ \mathbf{C}_k^w & \cdots & \mathbf{C}_k^w & \mathbf{C}_k^w \end{bmatrix} \tag{1.12}$$

of the joint process noise matrix with $\sum_{i=1}^{N} \omega_i = 1$ and $\omega_i \geq 0$ is used. The prediction of each local estimate $(\underline{\hat{x}}_k^{\mathrm{e},i}, \mathbf{C}_k^{\mathrm{e},i})$ at time k is then computed according to

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

and

$$\mathbf{C}_{k+1}^{\mathrm{p},i} = \mathbf{A}_k \mathbf{C}_k^{\mathrm{e},i} \mathbf{A}_k^{\mathrm{T}} + \tfrac{1}{\omega_i} \mathbf{C}_k^{w} \ . \tag{1.13}$$

Compared to the standard prediction formulas (1.3) and (1.4), the only difference lies in the inflated process noise matrix $\frac{1}{\omega_i}\mathbf{C}_k^w$. However, this modification allows to construct the cross-covariance terms

$$\mathrm{E}\left[(\mathbf{A}_k \, \underline{\hat{x}}_k^{\mathrm{e},i} - \underline{\boldsymbol{x}}_{k+1})(\mathbf{A}_k \, \underline{\hat{x}}_k^{\mathrm{e},j} - \underline{\boldsymbol{x}}_{k+1})^{\mathrm{T}}\right] = \begin{cases} \mathbf{A}_k \mathbf{C}_k^{\mathrm{e},i} \mathbf{A}_k^{\mathrm{T}} + \frac{1}{\omega_i} \mathbf{C}_k^{w} & , \ i = j \ , \\ \mathbf{0} & , \ i \neq j \ , \end{cases}$$

i.e., the errors of the predicted estimates are regarded as being uncorrelated. More precisely, by using the left-hand sides of the inequalities (1.11) and (1.12), the local estimation errors can be deemed to be independent. Hence, there is no need to store and reconstruct any dependencies between the local estimates.

The filtering of local measurements can simply be carried out by applying the standard Kalman filter formulas (1.5) and (1.6) locally. In the central node, a global estimate can be formed by combining[1] the local estimates according to

$$\underline{\hat{x}}_k^{\mathrm{fus}} = \mathbf{C}_k^{\mathrm{fus}} \sum_{i=1}^{N} (\mathbf{C}_k^{\mathrm{e},i})^{-1} \underline{\hat{x}}_k^{\mathrm{e},i} \tag{1.14}$$

and

$$\mathbf{C}_k^{\mathrm{fus}} = \left(\sum_{i=1}^{N} (\mathbf{C}_k^{\mathrm{e},i})^{-1} \right)^{-1} \ . \tag{1.15}$$

This fusion result is optimal in terms of the mean-squared error when only a single prediction and filtering step takes place before fusion. After several processing steps, the federated Kalman filter generally does not reach the quality of the schemes from Sec. 1.4, i.e., the federated Kalman filter is optimal when a fusion and reinitialization takes place after each time step and is otherwise suboptimal. With the bounds (1.11) and (1.12), conservative estimates are computed so that nodes may also fail and still a valid fusion result is attainable.

1.5.2 Optimal Distributed Kalman Filtering

The distributed Kalman filter [9, 15] can be characterized as a further development of the federated Kalman filter and provides the optimal estimation

[1] This combination corresponds to the optimal fusion formulas discussed in Sec. 1.6.1 in the case of independent estimates.

result after any number of time steps while the federated formulation is only optimal in special cases. Given the prior knowledge $(\hat{\underline{x}}_0^{\mathrm{p}}, \mathbf{C}_0^{\mathrm{p}})$, each local node can be initialized with

$$(\bar{\underline{x}}_0^{\mathrm{p},i}, \bar{\mathbf{C}}_0^{\mathrm{p}}) := (\hat{\underline{x}}_0^{\mathrm{p}}, N \cdot \mathbf{C}_0^{\mathrm{p}}) \;, \tag{1.16}$$

which resembles the inflation technique (1.11) for the federated Kalman filter when each factor is set to $\omega_i = \frac{1}{N}$, and N is the number of nodes in the network. The node index i in $\bar{\mathbf{C}}_0^{\mathrm{p}}$ is omitted as this matrix is the same for each node.

The prediction step performed by each node is also similar to the federated Kalman filter. For the system model (1.1) without input, each node processes its parameter set $(\bar{\underline{x}}_k^{\mathrm{e},i}, \bar{\mathbf{C}}_k^{\mathrm{e}})$ according to

$$\bar{\underline{x}}_{k+1}^{\mathrm{p},i} = \mathbf{A}_k \, \bar{\underline{x}}_k^{\mathrm{e},i} \tag{1.17}$$

and

$$\bar{\mathbf{C}}_{k+1}^{\mathrm{p}} = \mathbf{A}_k \bar{\mathbf{C}}_k^{\mathrm{e}} \mathbf{A}_k + N \cdot \mathbf{C}_k^{w} \;, \tag{1.18}$$

where $\bar{\mathbf{C}}_{k+1}^{\mathrm{p}}$ is the same for each node as long as they have the same $\bar{\mathbf{C}}_k^{\mathrm{e}}$. Furthermore, (1.18) is identical to the federated prediction step (1.13) for $\omega_i = \frac{1}{N}$. If the relationship

$$\hat{\underline{x}}_k^{\mathrm{e}} = \frac{1}{N} \sum_{i=1}^{N} \bar{\underline{x}}_k^{\mathrm{e},i} \quad \text{and} \quad \mathbf{C}_k^{\mathrm{e}} = \frac{1}{N} \bar{\mathbf{C}}_k^{\mathrm{e}} \tag{1.19}$$

holds for the local estimates $(\bar{\underline{x}}_k^{\mathrm{e},i}, \bar{\mathbf{C}}_k^{\mathrm{e}})$, i.e., (1.19) gives the centrally optimal estimate, then the same holds for the predicted results $(\bar{\underline{x}}_{k+1}^{\mathrm{p},i}, \bar{\mathbf{C}}_{k+1}^{\mathrm{p}})$.

The filtering step significantly differs from the federated Kalman filter. A locally available measurement $\hat{\underline{z}}_k^i$ at time k is combined with $\bar{\underline{x}}_{k+1}^{\mathrm{p},i}$ through

$$\bar{\underline{x}}_k^{\mathrm{e},i} = \bar{\mathbf{C}}_k^{\mathrm{e}} \left((\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1} \, \bar{\underline{x}}_k^{\mathrm{p},i} + (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \, \hat{\underline{z}}_k^i \right) \;, \tag{1.20}$$

which is similar to the standard update (1.5) with the inverse covariance gain (1.8). However, the difference lies in the matrix $\bar{\mathbf{C}}_k^{\mathrm{e}}$ which is computed by

$$\bar{\mathbf{C}}_k^{\mathrm{e}} = \left((\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1} + (\bar{\mathbf{C}}_k^{z})^{-1} \right)^{-1} \;, \tag{1.21}$$

where the globalized measurement covariance matrix

$$(\bar{\mathbf{C}}_k^{z})^{-1} = \frac{1}{N} \sum_{j=1}^{N} (\mathbf{H}_k^j)^{\mathrm{T}} (\mathbf{C}_k^{z,j})^{-1} \mathbf{H}_k^j \tag{1.22}$$

instead of only $(\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \mathbf{H}_k^i$ is used. More precisely, the matrix (1.22) incorporates the sensor parameters of all nodes in the network, i.e., all measurement and noise matrices. With this globalized covariance matrix, the local updates $(\bar{\underline{x}}_k^{\mathrm{e},i}, \bar{\mathbf{C}}_k^{\mathrm{e}})$ again share the same parameter $\bar{\mathbf{C}}_k^{\mathrm{e}}$. Of course, the vectors $\bar{\underline{x}}_k^{\mathrm{e},i}$ are different from each other due to the local measurement outcomes in (1.20). It is important to notice that none of the sets $(\bar{\underline{x}}_k^{\mathrm{e},i}, \bar{\mathbf{C}}_k^{\mathrm{e}})$ represents a valid state estimate, because both $\bar{\underline{x}}_k^{\mathrm{e},i}$ and $\bar{\mathbf{C}}_k^{\mathrm{e}}$ have been computed with the aid of the globalized matrix (1.21) instead of the matrices $(\mathbf{H}_k^i)^{\mathrm{T}}(\mathbf{C}_k^{z,i})^{-1}\mathbf{H}_k^i$, which refer only to the local measurement model.

The reason for the globalization becomes apparent when the fusion result at the center node is scrutinized. The fusion formulas are given by the invariant relation (1.19), which yields

$$\begin{aligned}(\mathbf{C}_k^{\mathrm{fus}})^{-1} = \left(\frac{1}{N}\bar{\mathbf{C}}_k^{\mathrm{e}}\right)^{-1} &= N(\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1} + N(\bar{\mathbf{C}}_k^{z})^{-1} \\ &= (\mathbf{C}_k^{\mathrm{p}})^{-1} + \sum_{j=1}^{N}(\mathbf{H}_k^j)^{\mathrm{T}}(\mathbf{C}_k^{z,j})^{-1}\mathbf{H}_k^j \ ,\end{aligned} \tag{1.23}$$

for the fused covariance matrix and

$$\begin{aligned}\hat{\underline{x}}_k^{\mathrm{fus}} &= \frac{1}{N}\sum_{i=1}^{N}\bar{\underline{x}}_k^{\mathrm{e},i} = \frac{1}{N}\sum_{i=1}^{N}\bar{\mathbf{C}}_k^{\mathrm{e}}\left((\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1}\,\bar{\underline{x}}_k^{\mathrm{p},i} + (\mathbf{H}_k^i)^{\mathrm{T}}(\mathbf{C}_k^{z,i})^{-1}\,\hat{\underline{z}}_k^i\right) \\ &= \mathbf{C}_k^{\mathrm{fus}}\sum_{i=1}^{N}(\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1}\,\bar{\underline{x}}_k^{\mathrm{p},i} + \mathbf{C}_k^{\mathrm{fus}}\sum_{i=1}^{N}(\mathbf{H}_k^i)^{\mathrm{T}}(\mathbf{C}_k^{z,i})^{-1}\,\hat{\underline{z}}_k^i \\ &= \mathbf{C}_k^{\mathrm{fus}}\,(\mathbf{C}_k^{\mathrm{p}})^{-1}\hat{\underline{x}}_k^{\mathrm{p}} + \mathbf{C}_k^{\mathrm{fus}}\sum_{i=1}^{N}(\mathbf{H}_k^i)^{\mathrm{T}}(\mathbf{C}_k^{z,i})^{-1}\,\hat{\underline{z}}_k^i\end{aligned} \tag{1.24}$$

for the fused estimate, where the invariant relation (1.19) has also been employed for $\bar{\underline{x}}_k^{\mathrm{p},i}$ and $\bar{\mathbf{C}}_k^{\mathrm{p}}$. These fusion formulas resemble the formulas (1.9) and (1.10) of the information filter. Hence, the optimal estimate is obtained as if all sensor data have been processed by a centralized Kalman filter. In contrast to the centralized concepts in Sec. 1.4, measurements can here be processed locally and need only to be transferred to the center unit when an estimate is requested. However, the globalization (1.21) clearly demonstrates that this approach may suffer from strict requirements regarding the local availability of knowledge about utilized models. Each sensor node must be in the position to compute the globalized covariance matrix (1.22) and therefore must be aware of the other nodes' sensor models, i.e., of the measurement matrices $\mathbf{H}_k^i$ and noise matrices $\mathbf{C}_k^{z,i}$ of each other node. In particular, this means that the optimal distributed Kalman filter is not robust to communication and node failures. The local parameters $(\bar{\underline{x}}_k^{\mathrm{e},i}, \bar{\mathbf{C}}_k^{\mathrm{e}})$ itself provide no information about the state, and they must be available to the center node in their entirety before an estimate of the state can be computed. This stands in contrast to the federated Kalman filter, which provides valid local estimates and is more robust.

1.5.3 Hypothesizing Kalman Filtering

The distributed Kalman filter suffers from the severe problem that any failure prevents an unbiased estimate to be computed. If any failure occurs and is not reported to every node, the correct globalized matrix (1.21) cannot be derived. As a further consequence, each incorrect globalized matrix affects subsequent processing and fusion steps, and a valid estimate cannot be obtained anymore. This problem can be addressed by correction matrices $\mathbf{\Delta}_k^{\mathrm{e},i}$ that are computed in line with the local parameters $(\underline{\bar{x}}_k^{\mathrm{e},i}, \bar{\mathbf{C}}_k^{\mathrm{e}})$ [26, 27]. With the aid of the corresponding correction matrix, each parameter $\underline{\bar{x}}_k^{\mathrm{e},i}$ can be transformed into an unbiased estimate

$$\underline{\tilde{x}}_k^{\mathrm{e}} = (\mathbf{\Delta}_k^{\mathrm{e},i})^{-1}\, \underline{\bar{x}}_k^{\mathrm{e},i} \tag{1.25}$$

with $\mathrm{E}[\underline{\tilde{x}}_k^{\mathrm{e}} - \underline{\boldsymbol{x}}_k] = \underline{0}$, i.e., the matrix $\mathbf{\Delta}_k^{\mathrm{e},i}$ effects a debiasing of $\underline{\bar{x}}_k^{\mathrm{e},i}$.

If the initialization of the local parameters is carried out by means of (1.16), the prior correction matrix at each node i is the identity $\mathbf{\Delta}_0^{\mathrm{p},i} = \mathbf{I}$. At a time step k, the prediction formulas (1.17) and (1.18) are accompanied with the update

$$\mathbf{\Delta}_{k+1}^{\mathrm{p},i} = \mathbf{A}_k\, \mathbf{\Delta}_k^{\mathrm{e},i} \mathbf{A}_k^{-1}$$

of the current correction matrix $\mathbf{\Delta}_k^{\mathrm{e},i}$, which gives the unbiased estimate

$$\underline{\tilde{x}}_{k+1}^{\mathrm{p}} = (\mathbf{\Delta}_{k+1}^{\mathrm{p},i})^{-1}\, \underline{\bar{x}}_{k+1}^{\mathrm{p},i} = \mathbf{A}_k\, (\mathbf{\Delta}_k^{\mathrm{e},i})^{-1}\, \underline{\bar{x}}_k^{\mathrm{e},i} = \mathbf{A}_k\, \underline{\tilde{x}}_k^{\mathrm{e}} \ .$$

This result complies with the prediction of the unbiased estimate (1.25) and, hence, is still unbiased.

In the filtering step, the corresponding correction matrix is given by

$$\mathbf{\Delta}_k^{\mathrm{e},i} = \bar{\mathbf{C}}_k^{\mathrm{e}} \Big((\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1}\, \mathbf{\Delta}_k^{\mathrm{p},i} + (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1}\, \mathbf{H}_k^i \Big) \ ,$$

where $\bar{\mathbf{C}}_k^{\mathrm{p}}$ is the predicted globalized matrix (1.18), and $\bar{\mathbf{C}}_k^{\mathrm{e}}$ is the updated globalized matrix (1.21). With these matrices, any local parameter $\underline{\bar{x}}_k^{\mathrm{e},i}$ can be turned into an unbiased estimate of the state, irrespective of whether the globalized matrix (1.22) has been computed correctly or failures have happened.

The correction matrices are of particular use when a fusion result is to be computed, but errors and node failures may have occurred during the local processing. Even missing data at the central node can be treated. Let $\mathcal{M} \subseteq \{1, \ldots, N\}$ contain the indices of those nodes whose data are available to the center node. There, the fused parameter

$$\underline{\bar{x}}_k^{\mathrm{fus}} = \frac{1}{N} \sum_{i \in \mathcal{M}} \underline{\bar{x}}_k^{\mathrm{e},i}$$

can be computed according to (1.24). Due to node failures or missing data, $\underline{\bar{x}}_k^{\mathrm{fus}}$ is potentially biased, but with the aid of the fused correction matrix

$$\mathbf{\Delta}_k^{\mathrm{fus}} = \frac{1}{N} \sum_{i \in \mathcal{M}} \mathbf{\Delta}_k^{\mathrm{e},i} \ ,$$

the unbiased state estimate

$$\underline{\tilde{x}}_k^{\text{fus}} = (\mathbf{\Delta}^{\text{fus}})^{-1}\, \underline{\bar{x}}_k^{\text{fus}}$$

is obtained.

The presented technique cannot only be used to make the distributed Kalman filter robust to node and communication failures, but can also be employed to replace the globalized covariance matrix (1.22) by a hypothesis on the global measurement capacity such that each node does not need to have access to each other node's sensor models. More precisely, the hypothesizing Kalman filtering scheme can be employed to assess the global measurement performance of the entire network without the need to assess the measurement quality of each node separately. Compared to the centralized Kalman filter from Sec. 1.4, the hypothesizing Kalman filter provides suboptimal estimates. If the hypothesis coincides with the actual globalized covariance matrix, the fusion result is even optimal.

1.5.4 Consensus Kalman Filtering

Consensus Kalman filtering [22, 23] has become a well-known concept for distributed state estimation and pursues a different strategy compared to the previously considered filtering schemes. A global estimate on the state is not computed at a central unit, but the nodes agree on a consensus estimate. This idea is justified by the perception that each sensor node provides an estimate that refers to the same state.

Essentially, two different possibilities have been studied to arrive at a consensus. Either a consensus on sensor data is aspired [22], or a consensus on estimates is computed [23]. It is an interesting fact to note that the local processing that is required for a consensus on sensor data bears considerable resemblance to the optimal distributed Kalman filter from Sec. 1.5.2. At the initialization step, each node is also provided with an inflated covariance matrix, i.e.,

$$(\underline{\bar{x}}_0^{\text{p},i}, \bar{\mathbf{C}}_0^{\text{p},i}) := (\underline{\hat{x}}_0^{\text{p}}, N \cdot \mathbf{C}_0^{\text{p}}) \ ,$$

where N is the number of nodes in the network. For every local estimate $(\underline{\bar{x}}_k^{\text{e},i}, \bar{\mathbf{C}}_k^{\text{e},i})$ at time k, the same number is used in prediction step

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

and

$$\bar{\mathbf{C}}_{k+1}^{\text{p},i} = \mathbf{A}_k \bar{\mathbf{C}}_k^{\text{e}} \mathbf{A}_k + N \cdot \mathbf{C}_k^w$$

in order to inflate the process noise matrix.

In the filtering step, the local prior information $(\underline{\bar{x}}_k^{\text{p},i}, \bar{\mathbf{C}}_k^{\text{p},i})$ is updated according to

$$\underline{\bar{x}}_k^{\text{e},i} = \bar{\mathbf{C}}_k^{\text{e}} \left((\bar{\mathbf{C}}_k^{\text{p}})^{-1}\, \underline{\bar{x}}_k^{\text{p},i} + \underline{\bar{z}}_k \right) \ , \tag{1.26}$$

and

$$\bar{\mathbf{C}}_k^{\mathrm{e}} = \left((\bar{\mathbf{C}}_k^{\mathrm{p}})^{-1} + (\bar{\mathbf{C}}_k^{z})^{-1} \right)^{-1} . \tag{1.27}$$

If the parameters $\bar{\underline{z}}_k$ and $\bar{\mathbf{C}}_k^z$ would be given by

$$\bar{\underline{z}}_k = \frac{1}{N} \sum_{i=j}^{N} (\mathbf{H}_k^j)^{\mathrm{T}} (\mathbf{C}_k^{z,j})^{-1} \, \hat{\underline{z}}_k^j \tag{1.28}$$

and

$$(\bar{\mathbf{C}}_k^z)^{-1} = \frac{1}{N} \sum_{j=1}^{N} (\mathbf{H}_k^j)^{\mathrm{T}} \, (\mathbf{C}_k^{z,j})^{-1} \, \mathbf{H}_k^j \ , \tag{1.29}$$

then (1.26) corresponds to the multisensor filtering formulas (1.9) while (1.29) overestimates the true error covariance matrix (1.10) by factor N. Hence, each local $\bar{\underline{x}}_k^{\mathrm{e},i}$ would yield the globally optimal estimate $\hat{\underline{x}}_k^{\mathrm{e}}$. Note that (1.29) complies with (1.22), but (1.28) differs from the measurement used in (1.20).

As the global parameters (1.28) and (1.29) are not available to each node, a consensus algorithm is used to approximate these parameters. Each node sends its sensor data to its neighboring nodes $\mathcal{N}(i) \subseteq \{1, \ldots, N\}$, where each local filtering step is accompanied by the consensus steps

$$\bar{\underline{z}}_k = \mathbf{W}_{ii} \, (\mathbf{H}_k^i)^{\mathrm{T}} (\mathbf{C}_k^{z,i})^{-1} \, \hat{\underline{z}}_k^i + \sum_{j \in \mathcal{N}(i)} \mathbf{W}_{ij} \, (\mathbf{H}_k^j)^{\mathrm{T}} (\mathbf{C}_k^{z,j})^{-1} \, \hat{\underline{z}}_k^j$$

and

$$(\bar{\mathbf{C}}_k^z)^{-1} = \mathbf{W}_{ii}^* \, (\mathbf{H}_k^i)^{\mathrm{T}} \, (\mathbf{C}_k^{z,i})^{-1} \, \mathbf{H}_k^i + \sum_{j \in \mathcal{N}(i)} \mathbf{W}_{ij}^* \, (\mathbf{H}_k^j)^{\mathrm{T}} \, (\mathbf{C}_k^{z,j})^{-1} \, \mathbf{H}_k^j \ .$$

These parameters then enter into equations (1.26) and (1.27). Many possibilities can be named to determine the weighting matrices $\mathbf{W}_{ij}$ and $\mathbf{W}_{ij}^*$. Examples are nearest neighboring or Metropolis weights [29].

Consensus Kalman filtering offers the advantage that no central processing unit is required, and an estimate can be retrieved from every node. In this regard, consensus filtering can also be viewed as a decentralized processing scheme. However, the focus lies on a cooperative solution of the estimation problem in form of synchronized estimates. The local performance is not optimized, and a potential disadvantage is the problem that, in general, the error covariance matrix (1.27) does neither represent the actual error nor is a valid bound.

1.6 Decentralized State Estimation and Data Fusion

Distributed Kalman filtering usually relies on nodes that closely cooperate with each other. Even a single failure may bring down the entire system such

TABLE 1.2
Overview of data fusion techniques.

method	**knowledge about dependencies**	**fusion result**
Bar-Shalom/Campo fusion	dependencies are entirely known	optimal
ellipsoidal intersection	unknown common sensor data	conservative
covariance intersection	dependencies are unknown	conservative

that no estimate can be provided, as in the optimal distributed scheme from Sec. 1.5.2. This stands in stark contrast to a decentralized processing. In decentralized estimation architectures, sensor nodes are essentially intended to operate independently of each other without being reliant on a central processing unit. They can share information with each other for the purpose of solving the estimation problem on a higher, cooperative level, but the primary goal is to optimize the local estimation performance.

In this section, it is assumed that each node is in the position to optimally process local sensor data according to the standard Kalman filter formulas from Sec. 1.2.1 and to provide an estimate on the state. In contrast to distributed schemes, local prediction and filtering steps generally do not need to be altered but are accompanied by data fusion as a third processing step. Estimates can be exchanged between nodes, and data fusion contributes to significantly improving the local estimation quality. Due to the independent local processing, dependencies between the estimates to be fused typically remain concealed. Even if dependencies are known and can be exploited, optimal fusion does not provide the same results as the centralized processing of measurements in Sec. 1.4.

The data fusion concepts discussed in the following subsections differ from one another in the amount of information that can be exploited. As discussed in Sec. 1.3, double-counting of sensor data and common process noise cause dependencies and have to be addressed. Either the dependency structure can be reconstructed or has to be bounded conservatively. Ignored dependencies or the erroneous assumption of independence may lead to biased fusion results. For each of the following concepts, two estimates $(\underline{\hat{x}}_\mathsf{A}, \mathbf{C}_\mathsf{A})$ and $(\underline{\hat{x}}_\mathsf{B}, \mathbf{C}_\mathsf{B})$ provided by sensor nodes $\mathsf{A}, \mathsf{B} \in \{1, \ldots, N\}$ are considered which are to be fused at node A. Table 1.2 gives an overview of the considered fusion strategies.

1.6.1 Optimal Fusion

The Bar-Shalom/Campo formulas [2] represent the optimal solution to the fusion problem and are well-known in multisensor tracking applications. These formulas can be applied when the cross-covariance matrix $\mathbf{C}_\mathsf{AB}$ is known or can be reconstructed. By means of the gain

$$\mathbf{K}^{\text{fus}} = (\mathbf{C}_{\mathsf{A}} - \mathbf{C}_{\mathsf{AB}})(\mathbf{C}_{\mathsf{A}} + \mathbf{C}_{\mathsf{B}} - \mathbf{C}_{\mathsf{AB}} - \mathbf{C}_{\mathsf{BA}})^{-1} \ ,$$

the fused estimate

$$\underline{\hat{x}}^{\text{fus}} = (\mathbf{I} - \mathbf{K}^{\text{fus}})\, \underline{\hat{x}}_{\mathsf{A}} + \mathbf{K}^{\text{fus}}\, \underline{\hat{x}}_{\mathsf{B}} \tag{1.30}$$

with error covariance matrix

$$\mathbf{C}^{\text{fus}} = \mathbf{C}_{\mathsf{A}} - \mathbf{K}^{\text{fus}}\,(\mathbf{C}_{\mathsf{A}} - \mathbf{C}_{\mathsf{BA}}) \tag{1.31}$$

can be computed. As mentioned before, the fusion result $(\underline{\hat{x}}^{\text{fus}}, \mathbf{C}^{\text{fus}})$ does not represent the optimal Kalman filter estimate given all the measurements that have been used to compute both estimates $\underline{\hat{x}}_{\mathsf{A}}$ and $\underline{\hat{x}}_{\mathsf{B}}$. The fusion result is optimal in a maximum-likelihood sense [6].

The Bar-Shalom/Campo formulas can be generalized to Millman's formulas [28] that allow for a simultaneous fusion of more than two estimates. The prerequisite of full knowledge of the cross-covariance terms constitutes a significant drawback of the optimal fusion technique since bookkeeping of the joint cross-covariance matrix is expensive in terms of both storage and processing requirements.

1.6.2 Ellipsoidal Intersection

In the situation that the estimates share common data and no other source of dependent information is present, a conservative fusion result can be achieved by means of the ellipsoidal intersection technique [29, 30]. In particular, the algorithm can be employed when no local prediction steps take place or the process noise is negligible such that the problem of common process noise does not occur.

The underlying idea is to remove the "maximum" possible common information from the fusion result. For this, the information form of the estimates is considered, i.e., $\underline{\hat{y}}_{\mathsf{A}} = \mathbf{Y}_{\mathsf{A}}\,\underline{\hat{x}}_{\mathsf{A}}$ and $\underline{\hat{y}}_{\mathsf{B}} = \mathbf{Y}_{\mathsf{B}}\,\underline{\hat{x}}_{\mathsf{B}}$ with $\mathbf{Y}_{\mathsf{A}} = \mathbf{C}_{\mathsf{A}}^{-1}$ and $\mathbf{Y}_{\mathsf{B}} = \mathbf{C}_{\mathsf{B}}^{-1}$, respectively. The assumption that only common data cause dependencies implies that each estimate can be decomposed into

$$\underline{\hat{y}}_{\mathsf{A}} = \underline{\hat{y}}_{\mathsf{A}}^{\mathrm{I}} + \underline{\bar{y}} \quad \text{and} \quad \mathbf{Y}_{\mathsf{A}} = \mathbf{Y}_{\mathsf{A}}^{\mathrm{I}} + \bar{\mathbf{Y}} \tag{1.32}$$

and

$$\underline{\hat{y}}_{\mathsf{B}} = \underline{\hat{y}}_{\mathsf{B}}^{\mathrm{I}} + \underline{\bar{y}} \quad \text{and} \quad \mathbf{Y}_{\mathsf{B}} = \mathbf{Y}_{\mathsf{B}}^{\mathrm{I}} + \bar{\mathbf{Y}} \tag{1.33}$$

where $\underline{\bar{y}}$ denotes the common information vector with information matrix $\bar{\mathbf{Y}}$. $\underline{\hat{y}}_{\mathsf{A}}^{\mathrm{I}}$ and $\underline{\hat{y}}_{\mathsf{B}}^{\mathrm{I}}$ represent information that is independent of each other. The optimal fusion result would then be obtained through

$$\underline{\hat{y}}^{\text{opt}} = \underline{\hat{y}}_{\mathsf{A}}^{\mathrm{I}} + \underline{\hat{y}}_{\mathsf{B}}^{\mathrm{I}} + \underline{\bar{y}} = \underline{\hat{y}}_{\mathsf{A}} + \underline{\hat{y}}_{\mathsf{B}} - \underline{\bar{y}}$$

and

$$\mathbf{Y}^{\text{opt}} = \mathbf{Y}_{\mathsf{A}}^{\text{I}} + \mathbf{Y}_{\mathsf{B}}^{\text{I}} + \bar{\mathbf{Y}} = \mathbf{Y}_{\mathsf{A}} + \mathbf{Y}_{\mathsf{B}} - \bar{\mathbf{Y}} \ ,$$

where on the right-hand side the parameters $\underline{\bar{y}}$ and $\bar{\mathbf{Y}}$ have to be removed so as to prevent double counting. As the common information $\underline{\bar{y}}$ and $\bar{\mathbf{Y}}$ is unknown, these terms have to be "maximized". From (1.32) and (1.33), the inequalities

$$\mathbf{Y}_{\mathsf{A}} \geq \bar{\mathbf{Y}} \quad \text{and} \quad \mathbf{Y}_{\mathsf{B}} \geq \bar{\mathbf{Y}}$$

can be deduced, which corresponds to the inequalities

$$\mathbf{C}_{\mathsf{A}} \leq \bar{\mathbf{C}}, \quad \mathbf{C}_{\mathsf{B}} \leq \bar{\mathbf{C}},$$

for the covariance matrices. $\bar{\mathbf{C}}$ is computed as the smallest covariance ellipsoid circumscribing the covariance ellipsoids of $\mathbf{C}_{\mathsf{A}}$ and $\mathbf{C}_{B}$, which is the Löwner-John ellipsoid. With the diagonalizations

$$\mathbf{C}_{\mathsf{A}} = \mathbf{Q}_{\mathsf{A}}\,\mathbf{D}_{\mathsf{A}}\,\mathbf{Q}_{\mathsf{A}}^{-1} \quad \text{and} \quad \mathbf{D}_{\mathsf{A}}^{-\frac{1}{2}}\,\mathbf{Q}_{\mathsf{A}}^{-1}\,\mathbf{C}_{\mathsf{B}}\,\mathbf{Q}_{\mathsf{A}}\,\mathbf{D}_{\mathsf{A}}^{-\frac{1}{2}} = \mathbf{Q}_{\mathsf{B}}\,\mathbf{D}_{\mathsf{B}}\,\mathbf{Q}_{\mathsf{B}}^{-1} \ ,$$

the smallest upper bound $\bar{\mathbf{C}}$ yields

$$\bar{\mathbf{C}} = \mathbf{Q}_{\mathsf{A}}\,\mathbf{D}_{\mathsf{A}}^{\frac{1}{2}}\,\mathbf{Q}_{\mathsf{B}}\,\bar{\mathbf{D}}\,\mathbf{Q}_{\mathsf{B}}^{-1}\,\mathbf{D}_{\mathsf{A}}^{\frac{1}{2}}\,\mathbf{Q}_{\mathsf{A}}^{-1} \ ,$$

where the diagonal matrix $\bar{\mathbf{D}}$ has the entries $(\bar{\mathbf{D}})_{ii} = \max\{1, (\mathbf{D}_{\mathsf{B}})_{ii}\}, i = 1, \ldots, n_x$. The resulting matrix $\bar{\mathbf{C}}$ can then be used to compute the conservative fusion result

$$\underline{\hat{x}}_{\text{EI}} = \mathbf{C}_{\text{EI}}\left(\mathbf{C}_{\mathsf{A}}^{-1}\,\underline{\hat{x}}_{\mathsf{A}} + \mathbf{C}_{\mathsf{B}}^{-1}\,\underline{\hat{x}}_{\mathsf{B}} - \bar{\mathbf{C}}^{-1}\,\underline{\bar{x}}\right)$$

and

$$\mathbf{C}_{\text{EI}} = \left(\mathbf{C}_{\mathsf{A}}^{-1} + \mathbf{C}_{\mathsf{B}}^{-1} - \bar{\mathbf{C}}^{-1}\right)^{-1}$$

with

$$\underline{\bar{x}} = \left(\mathbf{C}_{\mathsf{A}}^{-1} + \mathbf{C}_{\mathsf{B}}^{-1} - 2\bar{\mathbf{C}}^{-1}\right)^{-1}\left((\mathbf{C}_{\mathsf{B}}^{-1} - \bar{\mathbf{C}}^{-1})\underline{\hat{x}}_{\mathsf{A}} + (\mathbf{C}_{\mathsf{A}}^{-1} - \bar{\mathbf{C}}^{-1})\underline{\hat{x}}_{\mathsf{B}}\right) \ .$$

The last equation is prone to numerical instabilities since $\bar{\mathbf{C}}$ is a tight approximation. With a small $\gamma > 0$, the terms $\mathbf{C}_{\mathsf{A}}^{-1} - \bar{\mathbf{C}}^{-1} + \gamma\mathbf{I}$ and $\mathbf{C}_{\mathsf{B}}^{-1} - \bar{\mathbf{C}}^{-1} + \gamma\mathbf{I}$ become strictly positive definite.

1.6.3 Covariance Intersection

The covariance intersection algorithm [12, 13] has developed into the most important concept for decentralized data fusion. While the Bar-Shalom/Campo fusion formulas are reliant on a complete knowledge about the underlying dependencies, covariance intersection pursues the opposite direction and does not require any knowledge about the dependency structure. However, this advantage is paid with less informative estimation results. The fusion of the two estimates $(\underline{\hat{x}}_{\mathsf{A}}, \mathbf{C}_{\mathsf{A}})$ and $(\underline{\hat{x}}_{\mathsf{B}}, \mathbf{C}_{\mathsf{B}})$ is performed according to

$$\underline{\hat{x}}_{\mathsf{CI}} = \mathbf{C}_{\mathsf{CI}} \big(\omega\, \mathbf{C}_{\mathsf{A}}^{-1} \underline{\hat{x}}_{\mathsf{A}} + (1-\omega)\, \mathbf{C}_{\mathsf{B}}^{-1} \underline{\hat{x}}_{\mathsf{B}}\big) \tag{1.34}$$

for the fused estimate and

$$\mathbf{C}_{\mathsf{CI}} = \big(\omega\, \mathbf{C}_{\mathsf{A}}^{-1} + (1-\omega)\, \mathbf{C}_{\mathsf{B}}^{-1}\big)^{-1} \tag{1.35}$$

for the corresponding error covariance matrix. The weighting parameter $\omega \in [0,1]$ is usually determined to minimize the trace or determinant of the covariance matrix (1.35), which is a convex optimization problem. Also, approximate closed-form [19], information theoretic [11], and set-theoretic [25] solutions have been proposed. The CI algorithm yields covariance-consistent estimates with

$$\mathbf{C}_{\mathsf{CI}} \geq \mathrm{E}\left[(\underline{\hat{x}}_{\mathsf{CI}} - \underline{\boldsymbol{x}}_k)(\underline{\hat{x}}_{\mathsf{CI}} - \underline{\boldsymbol{x}}_k)^{\mathrm{T}}\right] ,$$

irrespective of the actual cross-covariance matrix $\mathbf{C}_{\mathsf{AB}} = \mathrm{E}[(\underline{\hat{x}}_{\mathsf{A}} - \underline{\boldsymbol{x}}_k)(\underline{\hat{x}}_{\mathsf{B}} - \underline{\boldsymbol{x}}_k)^{\mathrm{T}}]$ and choice of ω, provided that $(\underline{\hat{x}}_{\mathsf{A}}, \mathbf{C}_{\mathsf{A}})$ and $(\underline{\hat{x}}_{\mathsf{B}}, \mathbf{C}_{\mathsf{B}})$ are consistent estimates.

Covariance Bounds

The covariance intersection algorithm can also be derived in terms of upper bounds on the joint error covariance matrix. For the purpose of fusing two estimates $(\underline{\hat{x}}_{\mathsf{A}}, \mathbf{C}_{\mathsf{A}})$ and $(\underline{\hat{x}}_{\mathsf{B}}, \mathbf{C}_{\mathsf{B}})$, the covariance bounds algorithm [8, 24] provides the conservative approximation

$$\begin{bmatrix} \frac{1}{\omega}\mathbf{C}_{\mathsf{A}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega}\mathbf{C}_{\mathsf{B}} \end{bmatrix} \geq \begin{bmatrix} \mathbf{C}_{\mathsf{A}} & \mathbf{C}_{\mathsf{AB}} \\ \mathbf{C}_{\mathsf{BA}} & \mathbf{C}_{\mathsf{B}} \end{bmatrix} = \mathrm{E}\left[\left(\begin{bmatrix} \underline{\hat{x}}_{\mathsf{A}} \\ \underline{\hat{x}}_{\mathsf{B}} \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \underline{\boldsymbol{x}}\right)\left(\begin{bmatrix} \underline{\hat{x}}_{\mathsf{A}} \\ \underline{\hat{x}}_{\mathsf{B}} \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \underline{\boldsymbol{x}}\right)^{\mathrm{T}}\right]$$

of the joint error covariance matrix with $\omega \in (0,1)$. This bound again holds for every possible cross-covariance matrix $\mathbf{C}_{\mathsf{AB}}$. By employing the bound as the current joint MSE matrix, the estimates are deemed to be uncorrelated, and the estimates can be fused as if they are independent. More precisely, the estimates are replaced by $(\underline{\hat{x}}_{\mathsf{A}}, \frac{1}{\omega}\mathbf{C}_{\mathsf{A}})$ and $(\underline{\hat{x}}_{\mathsf{B}}, \frac{1}{1-\omega}\mathbf{C}_{\mathsf{B}})$ with inflated covariance matrices. Their fusion result directly yields (1.34) and (1.35).

Split Covariance Intersection

Partial knowledge about independent information can be incorporated into the covariance intersection algorithm. With split covariance intersection [13], independent parts of the local estimates can be exploited. For the local error covariance matrices

$$\mathbf{C}_{\mathsf{A}} = \mathbf{C}_{\mathsf{A}}^{\mathrm{I}} + \mathbf{C}_{\mathsf{A}}^{\mathrm{D}}$$

and

$$\mathbf{C}_{\mathsf{B}} = \mathbf{C}_{\mathsf{B}}^{\mathrm{I}} + \mathbf{C}_{\mathsf{B}}^{\mathrm{D}} \ ,$$

where $\mathbf{C}_{\mathsf{A}}^{\mathrm{I}}$ and $\mathbf{C}_{\mathsf{A}}^{\mathrm{I}}$ refer to parts that are known to be independent. The fusion result can then be computed according to

$$\underline{\hat{x}}_{\mathrm{sCI}} = \mathbf{C}_{\mathrm{sCI}}\Big(\omega\big(\omega\mathbf{C}_{\mathsf{A}}^{\mathrm{I}} + \mathbf{C}_{\mathsf{A}}^{\mathrm{D}}\big)^{-1}\underline{\hat{x}}_{\mathsf{A}} + (1-\omega)\big((1-\omega)\mathbf{C}_{\mathsf{B}}^{\mathrm{I}} + \mathbf{C}_{\mathsf{B}}^{\mathrm{D}}\big)^{-1}\underline{\hat{x}}_{\mathsf{B}}\Big)$$

and

$$\begin{aligned}\mathbf{C}_{\mathrm{sCI}}^{-1} &= \omega\Big(\omega\mathbf{C}_{\mathsf{A}}^{\mathrm{I}} + \mathbf{C}_{\mathsf{A}}^{\mathrm{D}}\Big)^{-1} + (1-\omega)\Big((1-\omega)\mathbf{C}_{\mathsf{B}}^{\mathrm{I}} + \mathbf{C}_{\mathsf{B}}^{\mathrm{D}}\Big)^{-1} \\ &= \Big(\mathbf{C}_{\mathsf{A}}^{\mathrm{I}} + \frac{1}{\omega}\mathbf{C}_{\mathsf{A}}^{\mathrm{D}}\Big)^{-1} + \Big(\mathbf{C}_{\mathsf{B}}^{\mathrm{I}} + \frac{1}{1-\omega}\mathbf{C}_{\mathsf{B}}^{\mathrm{D}}\Big)^{-1} \ ,\end{aligned}$$

where $\omega \in [0,1]$ is again chosen to fulfill some optimality criterion. In order to apply split covariance intersection, it is apparently not necessary to also split the according estimates $\underline{\hat{x}}_{\mathsf{A}}$ and $\underline{\hat{x}}_{\mathsf{B}}$ into dependent and independent parts. In terms of covariance bounds, split covariance intersection complies with the outer approximation

$$\begin{bmatrix}\mathbf{C}_{\mathsf{A}}^{\mathrm{I}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathsf{B}}^{\mathrm{I}}\end{bmatrix} + \begin{bmatrix}\frac{1}{\omega}\mathbf{C}_{\mathsf{A}}^{\mathrm{D}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega}\mathbf{C}_{\mathsf{B}}^{\mathrm{D}}\end{bmatrix} \geq \begin{bmatrix}\mathbf{C}_{\mathsf{A}}^{\mathrm{I}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathsf{B}}^{\mathrm{I}}\end{bmatrix} + \begin{bmatrix}\mathbf{C}_{\mathsf{A}}^{\mathrm{D}} & \mathbf{C}_{\mathsf{AB}}^{\mathrm{D}} \\ \mathbf{C}_{\mathsf{BA}}^{\mathrm{D}} & \mathbf{C}_{\mathsf{B}}^{\mathrm{D}}\end{bmatrix}$$

of the joint error covariance matrix, which is then utilized to fuse the estimates.

Covariance bounds also provide the means to exploit partial information [21] about cross-covariance terms $\mathbf{C}_{\mathsf{AB}}^{\mathrm{C}}$, i.e.,

$$\begin{bmatrix}\mathbf{C}_{\mathsf{A}}^{\mathrm{C}} & \mathbf{C}_{\mathsf{AB}}^{\mathrm{C}} \\ \mathbf{C}_{\mathsf{BA}}^{\mathrm{C}} & \mathbf{C}_{\mathsf{B}}^{\mathrm{C}}\end{bmatrix} + \begin{bmatrix}\frac{1}{\omega}\mathbf{C}_{\mathsf{A}}^{\mathrm{D}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega}\mathbf{C}_{\mathsf{B}}^{\mathrm{D}}\end{bmatrix} \geq \begin{bmatrix}\mathbf{C}_{\mathsf{A}}^{\mathrm{C}} & \mathbf{C}_{\mathsf{AB}}^{\mathrm{C}} \\ \mathbf{C}_{\mathsf{BA}}^{\mathrm{C}} & \mathbf{C}_{\mathsf{B}}^{\mathrm{C}}\end{bmatrix} + \begin{bmatrix}\mathbf{C}_{\mathsf{A}}^{\mathrm{D}} & \mathbf{C}_{\mathsf{AB}}^{\mathrm{D}} \\ \mathbf{C}_{\mathsf{BA}}^{\mathrm{D}} & \mathbf{C}_{\mathsf{B}}^{\mathrm{D}} \ ,\end{bmatrix}$$

where the superscript D again represents parts with unknown dependencies. In order to exploit the left-hand side for fusion, the estimates $(\underline{\hat{x}}_{\mathsf{A}}, \mathbf{C}_{\mathsf{A}}^{\mathrm{C}} + \frac{1}{\omega}\mathbf{C}_{\mathsf{A}}^{\mathrm{D}})$ and $(\underline{\hat{x}}_{\mathsf{B}}, \mathbf{C}_{\mathsf{B}}^{\mathrm{C}} + \frac{1}{\omega}\mathbf{C}_{\mathsf{B}}^{\mathrm{D}})$ are fused by means of the Bar-Shalom/Campo formulas (1.30) and (1.31) with the known cross-covariance term $\mathbf{C}_{\mathsf{AB}} = \mathbf{C}_{\mathsf{AB}}^{\mathrm{C}}$.

1.7 Conclusions

Kalman filtering in networked systems of sensor nodes is a multi-faceted problem. The design of state estimation concepts cannot be detached from the technical and operational aspects of a network but can be abstracted from concrete technical realizations of a network. The problems faced by designers of state estimation architectures can be boiled down to the proper treatment of dependent information shared by different nodes. Dependencies mainly arise from double-counting of sensor data and local state transition models that share the same process noise term, and their treatment is oriented towards the underlying estimation architecture.

Centralized, distributed, and decentralized Kalman filtering schemes have been studied. In a centralized estimation system, a single Kalman filter has access to all sensor data. The high transmission rates are costly in terms of communication and computation requirements, but offer the advantage that the conditional independence of measurement data can still be exploited. In this regard, the information form of the Kalman filter appears to be most appropriate. Distributed implementations pursue the goal to distribute the workload among the sensor nodes and to reduce the frequency of data transfers. A close-to-optimal estimation quality and robustness to failures are two objectives that cannot be achieved simultaneously, and different concepts to reach a compromise between these objectives have been discussed. Decentralized state estimation is concerned with the question of how to fuse Kalman filter estimates. The optimal choice of a fusion strategy depends upon the knowledge about the underlying dependency structure. Known dependencies between the estimates to be fused can be exploited while unknown dependencies have to be treated conservatively so as to avoid erroneous fusion results. The more knowledge is available, the more informative the fusion result will be.

Bibliography

[1] Yaakov Bar-Shalom and Leon Campo. On the Track-to-Track Correlation Problem. *IEEE Transactions on Automatic Control*, 26(2):571–572, April 1981.

[2] Yaakov Bar-Shalom and Leon Campo. The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance. *IEEE Transactions on Aerospace and Electronic Systems*, 22(6):803–805, November 1986.

[3] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[4] Neal A. Carlson. Federated Filter for Fault-tolerant Integrated Navigation Systems. In *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS'88)*, pages 110–119, Orlando, Florida, USA, 1988. Record 'Navigation into the 21st Century' (IEEE Cat. No.88CH2675-7).

[5] Neal A. Carlson. Federated Square Root Filter for Decentralized Parallel Processors. *IEEE Transactions on Aerospace and Electronic Systems*, 26:517–525, 1990.

[6] Kuo-Chu Chang, Rajat K. Saha, and Yaakov Bar-Shalom. On Optimal Track-to-Track Fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1271–1276, October 1997.

[7] Dietrich Fränken and Andreas Hüpper. Improved Fast Covariance Intersection for Distributed Data Fusion. In *Proceedings of the 8th International Conference on Information Fusion (Fusion 2005)*, Philadelphia, Pennsylvania, USA, July 2005.

[8] Uwe D. Hanebeck, Kai Briechle, and Joachim 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)*, pages 85–90, Baden–Baden, Germany, August 2001.

[9] Hamid R. Hashemipour, Sumit Roy, and Alan J. Laub. Decentralized Structures for Parallel Kalman Filtering. *IEEE Transactions on Automatic Control*, 33(1):88–94, January 1988.

[10] Mu Hua, Tim Bailey, Paul Thompson, and Hugh Durrant-Whyte. Decentralised Solutions to the Cooperative Multi-Platform Navigation Problem. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 47, pages 1433–1449, April 2011.

[11] Michael 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, July 2002.

[12] Simon J. Julier and Jeffrey K. Uhlmann. A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In *Proceedings of the IEEE American Control Conference (ACC 1997)*, volume 4, pages 2369–2373, Albuquerque, New Mexico, USA, June 1997.

[13] Simon J. Julier and Jeffrey K. Uhlmann. *Handbook of Multisensor Data Fusion: Theory and Practice*, chapter General Decentralized Data Fusion with Covariance Intersection, pages 319–343. CRC Press, 2nd ed. edition, 2009.

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

[15] Wolfgang 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, August 2008.

[16] Wolfgang Koch. On Accumulated State Densities with Applications to Out-of-Sequence Measurement Processing. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4):2766–2778, October 2011.

[17] Martin E. Liggins II, David L. Hall, and James Llinas, editors. *Handbook of Multisensor Data Fusion: Theory and Practice.* The Electrical Engineering and Applied Signal Processing Series. CRC Press, 2nd ed. edition, 2009.

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

[19] Wolfgang Niehsen. Information Fusion based on Fast Covariance Intersection Filtering. In *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*, Annapolis, Maryland, USA, July 2002.

[20] Benjamin Noack. *State Estimation for Distributed Systems with Stochastic and Set-membership Uncertainties.* Karlsruhe Series on Intelligent

Sensor-Actuator-Systems 14. KIT Scientific Publishing, Karlsruhe, Germany, 2013.

[21] Benjamin Noack, Marcus Baum, and Uwe D. Hanebeck. Automatic Exploitation of Independencies for Covariance Bounding in Fully Decentralized Estimation. In *Proceedings of the 18th IFAC World Congress (IFAC 2011)*, Milan, Italy, August 2011.

[22] Reza Olfati-Saber. Distributed Kalman Filter with Embedded Consensus Filters. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2005)*, pages 8179–8184, Sevilla, Spain, December 2005.

[23] Reza Olfati-Saber. Distributed Kalman Filtering for Sensor Networks. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, pages 5492–5498, New Orleans, Louisiana, USA, December 2007.

[24] Steven Reece and Stephen Roberts. Robust, Low-Bandwidth, Multi-Vehicle Mapping. In *Proceedings of the 8th International Conference on Information Fusion (Fusion 2005)*, volume 2, Philadelphia, Pennsylvania, USA, July 2005.

[25] Marc Reinhardt, Benjamin Noack, Marcus Baum, and Uwe D. Hanebeck. Analysis of Set-theoretic and Stochastic Models for Fusion under Unknown Correlations. In *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, July 2011.

[26] Marc Reinhardt, Benjamin Noack, and Uwe D. Hanebeck. On Optimal Distributed Kalman Filtering in Non-ideal Situations. In *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, July 2012.

[27] Marc Reinhardt, Benjamin Noack, and Uwe D. Hanebeck. The Hypothesizing Distributed Kalman Filter. In *Proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2012)*, Hamburg, Germany, September 2012.

[28] Vladimir Shin, Younghee Lee, and Tae-Sun Choi. Generalized Millman's Formula and Its Application for Estimation Problems. *Signal Processing*, 86(2):257–266, 2006.

[29] Joris Sijs. *State Estimation in Networked Systems*. Ph.D. thesis, Technische Universiteit Eindhoven, April 2012.

[30] Joris Sijs, Mircea Lazar, and Paul P. J. van den Bosch. State-Fusion with Unknown Correlation: Ellipsoidal Intersection. In *Proceedings of the 2010 American Control Conference (ACC 2010)*, Baltimore, Maryland, USA, June 2010.

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