

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Distributed Kalman Filter

Felix Govaers

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.71941>

Abstract

The continuing trend toward connected sensors (“internet of things” and “ubiquitous computing”) drives a demand for powerful distributed estimation methodologies. In tracking applications, the distributed Kalman filter (DKF) provides an optimal solution under Kalman filter conditions. The optimal solution in terms of the estimation accuracy is also achieved by a centralized fusion algorithm, which receives all associated measurements. However, the centralized approach requires full communication of all measurements at each time step, whereas the DKF works at arbitrary communication rates since the calculation is fully distributed. A more recent methodology is based on “accumulated state density” (ASD), which augments the states from multiple time instants to overcome spatial cross-correlations. This chapter explains the challenges in distributed tracking. Then, possible solutions are derived, which include the DKF and ASD approach.

Keywords: distributed Kalman filter, target tracking, multisensor fusion, information fusion, accumulated state density

1. Introduction

Modern tracking and surveillance system development is increasingly driving technological trends and algorithm developments toward networks of dislocated sensors. Besides classical target tracking, many other applications can be found, for instance, in robotics, manufacturing, health care, and financial economics. A multisensor network can exploit spatial diversity to compensate for the weak attributes of a single sensor such as limited field of view or high measurement noise. Also, heterogeneous sensors can reveal a more complete situational awareness and a more precise estimate of the underlying phenomena. Additionally, a sensor network is more robust against a single point of failure in comparison to a standalone system, if its architecture is chosen carefully.

Despite its unquestioned advantages, a multisensor network must cope with design-specific challenges, for instance, a full transmission of all the measurements to a fusion center can be

hindered, when the communication links are unreliable or constrained in bandwidth or costs. A well-known approach to cope with the limited bandwidth data links is to apply data processing on the sensor sites to generate local track parameters that are transmitted to the fusion center. The latter then reconstructs the global track parameters by an application of a *Track-to-Track Fusion* (T2TF) scheme. Depending on the scenario constraints, this is a nontrivial task, since the local tracks are mutually correlated due to the common process noise. The first known solution in the literature to the T2TF problem proposed to apply an information filter-based multisensor fusion algorithm in [1], which later became famous as the “tracklet fusion.” However, the tracklet fusion also requires a transmission from each sensor after each time step in order to reconstruct the optimal solution.

This chapter presents the theory and the derivation of the *distributed Kalman filter* (DKF), which is an optimal solution of the T2TF problem under Kalman filter assumptions with respect to the *mean squared error* (MSE). Assuming Kalman conditions means that linear Gaussian models are provided for the motion model and all measurement models of the sensors. Moreover, it is assumed that measurement-to-track (at the sensors) and track-to-track (at the fusion center) association has been solved. The DKF approach requires, however, all remote sensor models to be known at each local sensor site, which is infeasible in most practical scenarios. Therefore, this chapter also presents a solution based on the *accumulated state density* (ASD), which is closely related to the DKF but does not require the measurement models to be known. Surveys that reflect the history of research in distributed estimation can be found, for instance, in [2, 3].

This chapter is structured as follows: Section 2 summarizes the problem formulation. A basic approach to T2TF is given in Section 3, where we present the least squares solution. Section 4 presents a simple fusion methodology, which is easy to compute and provides practical results for various applications. The reason why this approach is suboptimal is investigated in Section 5 by means of a recursive computation of the cross-covariances of the local tracks. In Section 6, the product representation of Gaussian probability densities is introduced, which is the basis for the derivation of the distributed Kalman filter in Section 7. An alternative derivation in terms of information parameters is provided in Section 8. Since the local measurement models are often unknown in practical applications, the distributed accumulated state density filter is introduced in Section 9. The chapter concludes with Section 10.

2. Problem formulation

Throughout this chapter, it is assumed that all S sensors produce their measurements $z_k^s \in \mathbb{R}^m$ at each time step t_k of the same target with its true state $x_k \in \mathbb{R}^n$ in a synchronized way. The extension to the unsynchronized case is straightforward by means of standard Kalman filter predictions, and is therefore omitted for the sake of notational simplicity. The measurement equation for sensor $s \in \{1, \dots, S\}$ is given by

$$z_k^s = H_k^s x_k + v_k^s \quad (1)$$

where $v_k^s \sim N(v_k^s; 0, R_k^s)$ is the Gaussian distributed, zero-mean random variable, which models the noise of the sensing process. Therefore, the likelihood for a single measurement is fully described by the Gaussian

$$p(z_k^s | x_k) = N(z_k^s; H_k^s x_k, R_k^s) \quad (2)$$

Since the measurement processes across all sensors $Z_k = \{z_k^1, \dots, z_k^S\}$ are mutually independent, the joint likelihood of all sensor data produced at time t_k factorizes:

$$\begin{aligned} p(Z_k | x_k) &= \prod_{s=1}^S p(z_k^s | x_k) \\ &= \prod_{s=1}^S N(z_k^s; H_k^s x_k, R_k^s) \end{aligned} \quad (3)$$

The true state of the target itself is modeled as a time-variant stochastic process, where the transition from time t_{k-1} to time t_k is given by the following motion equation:

$$x_k = F_{k|k-1} x_{k-1} + w_k \quad (4)$$

where $w_k \sim N(w_k; 0, Q_k)$ is the Gaussian distributed, zero-mean random variable to model the process noise of the system. Analogously to the likelihood, this provides the probability density function for the *transition model*:

$$p(x_k | x_{k-1}) = N(x_k; F_{k|k-1} x_{k-1}, Q_k) \quad (5)$$

Based on the local processors, each sensor node produces a *track* at time t_k in terms of an estimate $x_{k|k}^s$ and a corresponding estimation error covariance $P_{k|k}^s$. In a T2TF scheme, these parameters are the only information, which is transmitted to a *fusion center* (FC). It should be noted that the FC may also not be centralized, distinguished instance in the architecture, but each and every processing node can act as a FC. The introduction of a distinguished FC is only for clarification of different computation layers. An excellent overview of pros and cons of various data fusion layers can be found in [4].

The T2TF problem can now be stated as follows: compute a fused estimate $x_{k|k}$ of the state x_k and a consistent error covariance $P_{k|k}$ by means of the local tracks and knowledge on their models:

$$x_{k|k} \leftarrow \{x_{k|k}^1, \dots, x_{k|k}^S\} \quad (6)$$

3. Least squares estimate

In order to compute an estimate as a well-suited combination of the local tracks, it is useful to consider the joint likelihood function given by the following Gaussian:

$$p(x_{k|k}^1, \dots, x_{k|k}^S | x_k) = N \left(\begin{pmatrix} x_{k|k}^1 \\ \vdots \\ x_{k|k}^S \end{pmatrix}; \begin{pmatrix} x_k \\ \vdots \\ x_k \end{pmatrix}, \begin{pmatrix} P_{k|k}^{1,1} & \dots & P_{k|k}^{1,S} \\ \vdots & \ddots & \vdots \\ P_{k|k}^{S,1} & \dots & P_{k|k}^{S,S} \end{pmatrix} \right), \quad (7)$$

where $P_{k|k}^{s,s} = P_{k|k}^s$ are the track covariances on the block-diagonal entries and $P_{k|k}^{i,j} \triangleq \text{cov}[x_{k|k}^i, x_{k|k}^j | x_k] = P_{k|k}^{j,i T}$ are the cross-covariances on the off-diagonal entries of the joint error covariance.

Since the joint likelihood from above is proportional to an exponential function:

$$p(x_{k|k}^1, \dots, x_{k|k}^S | x_k) \propto \exp \left\{ -\frac{1}{2} \left(\begin{pmatrix} x_{k|k}^1 \\ \vdots \\ x_{k|k}^S \end{pmatrix} - \begin{pmatrix} x_k \\ \vdots \\ x_k \end{pmatrix} \right)^T \begin{pmatrix} P_{k|k}^{1,1} & \dots & P_{k|k}^{1,S} \\ \vdots & \ddots & \vdots \\ P_{k|k}^{S,1} & \dots & P_{k|k}^{S,S} \end{pmatrix}^{-1} \left(\begin{pmatrix} x_{k|k}^1 \\ \vdots \\ x_{k|k}^S \end{pmatrix} - \begin{pmatrix} x_k \\ \vdots \\ x_k \end{pmatrix} \right) \right\} \quad (8)$$

the *maximum likelihood* (ML) estimate can be computed in terms of the least squares:

$$x_{k|k} = \underset{x_k}{\text{argmin}} \left\{ \left(x_{k|k}^{1:S} - I^{1:S} x_k \right)^T \left(P_{k|k}^{1:S} \right)^{-1} \left(x_{k|k}^{1:S} - I^{1:S} x_k \right) \right\}, \quad (9)$$

where $x_{k|k}^{1:S} = (x_{k|k}^{1:T}, \dots, x_{k|k}^{S:T})^T$, $I^{1:S} = (I, \dots, I)^T$, and $P_{k|k}^{1:S}$ for the joint error covariance have been introduced for notational simplicity. A closed form solution of the ML estimates can be obtained by setting the gradient with respect to the state to zero:

$$0 = \nabla_{x_k} \left(x_{k|k}^{1:S} - I^{1:S} x_k \right)^T \left(P_{k|k}^{1:S} \right)^{-1} \left(x_{k|k}^{1:S} - I^{1:S} x_k \right) = 2(I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} \left(x_{k|k}^{1:S} - I^{1:S} x_k \right), \quad (10)$$

Therefore, the ML estimate is given by:

$$x_{k|k} = \left((I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} (I^{1:S}) \right)^{-1} (I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} x_{k|k}^{1:S}. \quad (11)$$

For information fusion applications, it is also important to have a consistent estimate of the squared error, in other words, we need to compute the corresponding error covariance:

$$\begin{aligned} \text{cov}[x_{k|k} | x_k] &= E \left[(x_k - x_{k|k})(x_k - x_{k|k})^T \right] \\ &\triangleq E \left[(x_k - x_{k|k})^2 \right] \\ &= E \left[\left(x_k - \left((I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} (I^{1:S}) \right)^{-1} (I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} x_{k|k}^{1:S} \right)^2 \right] \\ &= E \left[\left(\left((I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} (I^{1:S}) \right)^{-1} \left(\left((I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} (I^{1:S}) \right) x_k - (I^{1:S})^T \left(P_{k|k}^{1:S} \right)^{-1} x_{k|k}^{1:S} \right) \right)^2 \right] \end{aligned} \quad (12)$$

$$\begin{aligned}
&= \left((I^{1:S})^T (P_{k|k}^{1:S})^{-1} (I^{1:S}) \right)^{-1} (I^{1:S})^T (P_{k|k}^{1:S})^{-1} \mathbb{E} \left[\left((I^{1:S}) x_k - x_{k|k}^{1:S} \right)^2 \right] (P_{k|k}^{1:S})^{-1} (I^{1:S}) \left((I^{1:S})^T (P_{k|k}^{1:S})^{-1} (I^{1:S}) \right)^{-1} \\
&= \left((I^{1:S})^T (P_{k|k}^{1:S})^{-1} (I^{1:S}) \right)^{-1}.
\end{aligned}$$

The last equation holds due to the fact that $\mathbb{E} \left[\left((I^{1:S}) x_k - x_{k|k}^{1:S} \right)^2 \right] = P_{k|k}^{1:S}$ is the joint covariance. Concluding the derivations from above equation, one can obtain:

$$\begin{aligned}
x_{k|k} &= P_{k|k} (I^{1:S})^T (P_{k|k}^{1:S})^{-1} x_{k|k}^{1:S}, \\
P_{k|k} &= \left((I^{1:S})^T (P_{k|k}^{1:S})^{-1} (I^{1:S}) \right)^{-1}
\end{aligned} \tag{13}$$

4. Naïve fusion

It is obvious that for the ML estimate, it is assumed that the cross-covariances $P_{k|k}^{i,j}$, $i, j \in \{1, \dots, S\}$ are known. Since this might not be given in practical scenarios, a simple approximation is to assume them to be zero. This approach is called *Naïve fusion*. It implies that the joint error covariance is given in block-diagonal form:

$$P_{k|k}^{1:S} = \begin{pmatrix} P_{k|k}^{1,1} & & \\ & \ddots & \\ & & P_{k|k}^{S,S} \end{pmatrix} \tag{14}$$

As a direct consequence of the *matrix inversion lemma* (see Appendix 12.1), the inverse can be obtained in closed form solution:

$$\left(P_{k|k}^{1:S} \right)^{-1} = \begin{pmatrix} \left(P_{k|k}^{1,1} \right)^{-1} & & \\ & \ddots & \\ & & \left(P_{k|k}^{S,S} \right)^{-1} \end{pmatrix} \tag{15}$$

Filling into the maximum likelihood formulas directly yields.

$$\begin{aligned}
x_{k|k} &= P_{k|k} \sum_{s=1}^S \left(P_{k|k}^s \right)^{-1} x_{k|k}^s, \\
P_{k|k} &= \left(\sum_{s=1}^S \left(P_{k|k}^s \right)^{-1} \right)^{-1}
\end{aligned} \tag{16}$$

Thus, by means of a simple approximation of the ML estimate, we have obtained a first practical fusion rule for the FC, which we call *convex combination* due to its structure for further

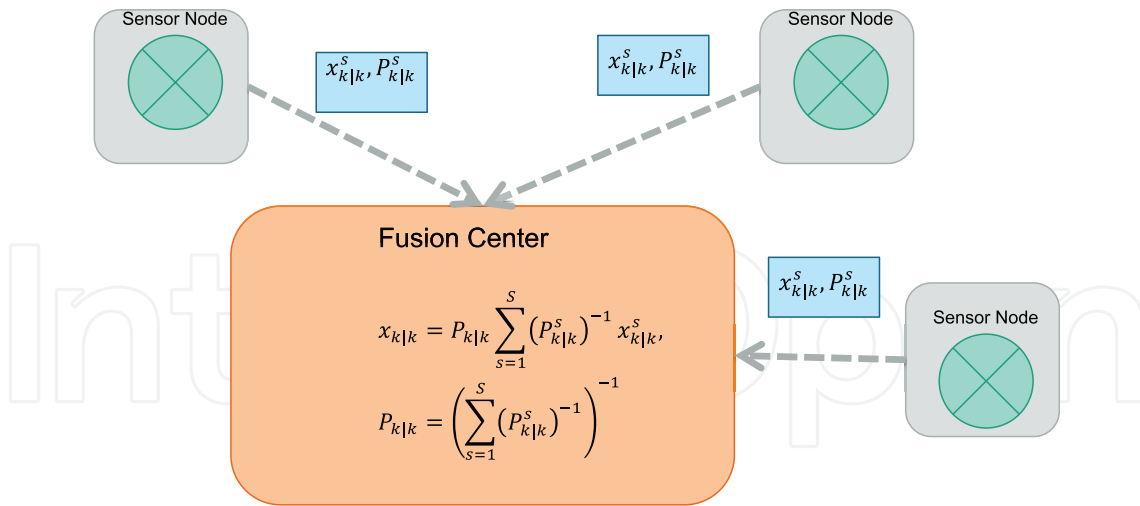


Figure 1. Fusion scheme of the Naive Fusion approach.

usage. The fusion scheme as a whole can be outlined schematically as in the flowing **Figure 1**. Each sensor node s processes its produced sensor data by means of a local filter, which results in a track in terms of an estimate together with an error covariance. These parameters are transmitted to the fusion center, which applies the convex combination to compute the fused result.

5. What makes the Naïve fusion naïve?

For the Naïve fusion, we have assumed that the cross-covariances vanish. It is worth to be aware of the structure of the cross-covariances to see the conditions whether this holds or does not hold. This can be achieved by a recursive computation of the posterior cross-covariance $P_{k|k}^{i,j}$ of two sensors with indices i and j , which process their data by means of local Kalman filters. At the beginning of the estimation process at t_0 tracks are not yet correlated, that is, $P_{0|0}^{i,j} = 0$, due to the fact that initial measurements are mutually uncorrelated. A recursive computation can be achieved by a *prediction-filtering* cycle.

5.1. Cross-covariance prediction

For the prediction step, it is assumed that a previous posterior cross-covariance $P_{k|k-1}^{i,j}$ has been computed. The prior parameters are obtained by means of the motion model:

$$\begin{aligned}
 P_{k|k-1}^{i,j} &= \mathbb{E} \left[\left(x_k - x_{k|k-1}^i \right) \left(x_k - x_{k|k-1}^j \right)^T \right] \\
 &= \mathbb{E} \left[\left(F_{k|k-1} x_{k-1} + w_k - F_{k|k-1} x_{k-1|k-1}^i \right) \left(F_{k|k-1} x_{k-1} + w_k - F_{k|k-1} x_{k-1|k-1}^j \right)^T \right] \quad (17) \\
 &= F_{k|k-1} \mathbb{E} \left[\left(x_{k-1} - x_{k-1|k-1}^i \right) \left(x_{k-1} - x_{k-1|k-1}^j \right)^T \right] F_{k|k-1}^T
 \end{aligned}$$

$$\begin{aligned} & -F_{k|k-1} \mathbb{E} \left[\left(x_{k-1} - x_{k-1|k-1}^i \right) w_k^T \right] - \mathbb{E} \left[w_k \left(x_{k-1} - x_{k-1|k-1}^j \right)^T \right] F_{k|k-1}^T + \mathbb{E} [w_k w_k^T] \\ & = F_{k|k-1} P_{k-1|k-1}^{i,j} F_{k|k-1}^T + Q_k, \end{aligned}$$

where the last equality holds due to the fact that the estimation errors at time t_{k-1} of both sensor processors are uncorrelated to the process noise w_k .

5.2. Cross-covariance filtering

In the filtering step, both sensors compute their posterior parameters based on the produced measurements z_k^i and z_k^j , respectively. It is assumed that the local processors have applied local Kalman filter update steps with Kalman gains $W_{k|k-1}^i$ and $W_{k|k-1}^j$. The cross-covariance posterior matrix can be obtained by a straightforward computation:

$$\begin{aligned} P_{k|k}^{i,j} &= \mathbb{E} \left[\left(x_k - x_{k|k}^i \right) \left(x_k - x_{k|k}^j \right)^T \right] \\ &= \mathbb{E} \left[\left(x_k - x_{k|k-1}^i - W_{k|k-1}^i \left(z_k^i - H_k^i x_{k|k-1}^i \right) \right) \left(x_k - x_{k|k-1}^j - W_{k|k-1}^j \left(z_k^j - H_k^j x_{k|k-1}^j \right) \right)^T \right] \\ &= \mathbb{E} \left[\left(x_k - x_{k|k-1}^i - W_{k|k-1}^i \left(H_k^i x_k + v_k^i - H_k^i x_{k|k-1}^i \right) \right) \left(x_k - x_{k|k-1}^j - W_{k|k-1}^j \left(H_k^j x_k + v_k^j - H_k^j x_{k|k-1}^j \right) \right)^T \right] \\ &= \mathbb{E} \left[\left(\left(I - W_{k|k-1}^i H_k^i \right) \left(x_k - x_{k|k-1}^i \right) - W_{k|k-1}^i v_k^i \right) \left(\left(I - W_{k|k-1}^j H_k^j \right) \left(x_k - x_{k|k-1}^j \right) - W_{k|k-1}^j v_k^j \right)^T \right] \\ &= \left(I - W_{k|k-1}^i H_k^i \right) \mathbb{E} \left[\left(x_k - x_{k|k-1}^i \right) \left(x_k - x_{k|k-1}^j \right)^T \right] \left(I - W_{k|k-1}^j H_k^j \right)^T + W_{k|k-1}^i \mathbb{E} [v_k^i v_k^{jT}] W_{k|k-1}^j{}^T \\ &= \left(I - W_{k|k-1}^i H_k^i \right) P_{k|k-1}^{i,j} \left(I - W_{k|k-1}^j H_k^j \right)^T \end{aligned} \tag{18}$$

For these equations, we have used the fact that the prior estimate error $\left(x_k - x_{k|k-1}^i \right)$ is independent of the measurement noise v_k^i , and that v_k^i and v_k^j are mutually independent.

Concluding the calculations from this section, we have obtained a recursive solution for the cross-covariances:

$$\begin{aligned} P_{0|0}^{i,j} &= 0 \\ P_{k|k-1}^{i,j} &= F_{k|k-1} P_{k-1|k-1}^{i,j} F_{k|k-1}^T + Q_k \\ P_{k|k}^{i,j} &= \left(I - W_{k|k-1}^i H_k^i \right) P_{k|k-1}^{i,j} \left(I - W_{k|k-1}^j H_k^j \right)^T \end{aligned} \tag{19}$$

One can see that the cross-covariances are zero, if and only if the process noise covariance Q_k vanishes. In other words, if the tracks refer to a deterministically moving target and all sensors do local Kalman filtering, then the convex combination equations yield the optimal fusion

result. If this is not the case, as maybe in most practical applications, then the Naïve fusion method is an approximation and its degree of approximation depends primarily on the level of the process noise.

6. Gaussian product representation

The basic concept of the distributed Kalman filter is to make the local parameters stochastically independent, even if process noise is present. This is achieved by a *product representation*, which directly follows from the fact that.

$$\begin{aligned} & \exp \left\{ -\frac{1}{2} \left(\begin{pmatrix} x_{k|k}^1 \\ \vdots \\ x_{k|k}^S \end{pmatrix} - \begin{pmatrix} x_k \\ \vdots \\ x_k \end{pmatrix} \right)^T \begin{pmatrix} (P_{k|k}^1)^{-1} & & \\ & \ddots & \\ & & (P_{k|k}^S)^{-1} \end{pmatrix} \left(\begin{pmatrix} x_{k|k}^1 \\ \vdots \\ x_{k|k}^S \end{pmatrix} - \begin{pmatrix} x_k \\ \vdots \\ x_k \end{pmatrix} \right) \right\} \\ &= \exp \left\{ -\frac{1}{2} \sum_{s=1}^S (x_{k|k}^s - x_k)^T (P_{k|k}^s)^{-1} (x_{k|k}^s - x_k) \right\} \propto \prod_{s=1}^S N(x_k; x_{k|k}^s, P_{k|k}^s) \end{aligned} \quad (20)$$

Thus, the Gaussian product representation is equivalent to uncorrelated track parameters for each processing node. It should be noted that the product representation is not normalized, that is, the integral for $S > 1$ is not unity. This, however, is not of practical relevance since the fused estimate density is a Gaussian and the parameters of which are provided by the convex combination.

7. Derivation of the distributed Kalman filter

For the DKF, we are going to modify the local processing scheme for each sensor in order to have the product representation hold at each instant of time. Then, when the fusion center receives the parameters from all sensors, the convex combination can be applied to compute the optimal global estimate. Note that the convex combination does not consider a local prior of the fusion center; therefore, the result will be independent from previous transmissions. This can be of great benefit, if communication channels with unreliable links have to be considered, since the full information on the target state is distributed in the sensor network. However, for completeness, it should also be noted that the modified local parameters are not optimal anymore in a local sense. One could say that local optimality is given up for the sake of global optimality [5].

In the following sections, the derivation of a prediction-filtering recursion of the DKF is discussed.

7.1. DKF prediction

For the prediction, it is assumed that the previous posterior is given in product representation:

$$p(x_{k-1}|Z^{k-1}) \propto \prod_{s=1}^S N(x_{k-1}; x_{k-1|k-1}^s, P_{k-1|k-1}^s) \quad (21)$$

For notational simplicity, we have conditioned the posterior on the full data set $Z^{k-1} = \{Z_1^{k-1}, \dots, Z_s^{k-1}\}$, where $Z_s^{k-1} \{Z_1^s, \dots, Z_{k-1}^s\}$ are from all sensors and all time steps up to time t_{k-1} , of which the local tracks are sufficient statistics. At the initialization phase, that is, $k-1=0$, the product representation is trivial, since the initial estimates can be based on first measurements, which are mutually independent.

To derive a closed form solution for the prediction of product representation, it is required to *globalize* the covariances $P_{k-1|k-1}^s$ of the local processing nodes at first. This process changes the local parameters such that the same previous posterior density is factorized; however, the local covariances will be unified to a single $\tilde{P}_{k-1|k-1}$. Rigorously speaking, this matrix does not represent a meaningful covariance in the sense of an expected estimation error squared anymore. Still, the fused result will be optimal since the global density is not changed during this process. Thus, if we set

$$p(x_{k-1}|Z^{k-1}) \propto \prod_{s=1}^S N(x_{k-1}; \tilde{x}_{k-1|k-1}^s, \tilde{P}_{k-1|k-1}) \quad (22)$$

where

$$\begin{aligned} \tilde{x}_{k-1|k-1}^s &= SP_{k-1|k-1} \left(P_{k-1|k-1}^s \right)^{-1} x_{k-1|k-1}^s \\ \tilde{P}_{k-1|k-1} &= SP_{k-1|k-1} \\ P_{k-1|k-1} &= \left(\sum_{s=1}^S \left(P_{k-1|k-1}^s \right)^{-1} \right)^{-1} \end{aligned} \quad (23)$$

then the same fused density will be obtained, which is easily verified by means of the convex combination. It should be noted that the remote error covariances $P_{k-1|k-1}^s$ are required to compute the globalized covariance matrix $\tilde{P}_{k-1|k-1}$. Since Kalman filter conditions are assumed, they can be computed without data transmission, as they do not depend on the local sensor measurements. Therefore, it is sufficient to be aware of the remote sensor models.

The prediction formulas can now be obtained by a marginalization of the joint density of the current and the last time step:

$$\begin{aligned}
p(x_k|Z^{k-1}) &= \int dx_{k-1} p(x_k, x_{k-1}|Z^{k-1}) \\
&= \int dx_{k-1} p(x_k|x_{k-1}, Z^{k-1}) p(x_{k-1}|Z^{k-1}) \\
&= \int dx_{k-1} p(x_k|x_{k-1}) p(x_{k-1}|Z^{k-1})
\end{aligned} \tag{24}$$

The last equality holds due to the Markov property of the system. Filling in our linear Gaussian transition model and the previous posterior yields

$$p(x_k|Z^{k-1}) \propto \int dx_{k-1} N(x_k; F_{k|k-1}x_{k-1}, Q_k) \prod_{s=1}^S N(x_{k-1}; \tilde{x}_{k-1|k-1}^s, \tilde{P}_{k-1|k-1}) \tag{25}$$

By means of a simple algebraic manipulation, it is possible to factorize the transition kernel Gaussian up to proportionality:

$$\begin{aligned}
N(x_k; F_{k|k-1}x_{k-1}, Q_k) &\propto \exp \left\{ -\frac{1}{2} (F_{k|k-1}x_{k-1} - x_k)^T (Q_k)^{-1} (F_{k|k-1}x_{k-1} - x_k) \right\} \\
&= \exp \left\{ -\frac{1}{2} (F_{k|k-1}x_{k-1} - x_k)^T S(SQ_k)^{-1} (F_{k|k-1}x_{k-1} - x_k) \right\} \\
&\propto [N(x_k; F_{k|k-1}x_{k-1}, SQ_k)]^S
\end{aligned} \tag{26}$$

Thus, we can factorize the integration term of the global prior completely:

$$p(x_k|Z^{k-1}) \propto \int dx_{k-1} \prod_{s=1}^S N(x_k; F_{k|k-1}x_{k-1}, SQ_k) N(x_{k-1}; \tilde{x}_{k-1|k-1}^s, \tilde{P}_{k-1|k-1}) \tag{27}$$

An application of the product formula (Section 12.2 in the appendix) yields:

$$p(x_k|Z^{k-1}) \propto \prod_{s=1}^S N(x_k; x_{k|k-1}^s, \tilde{P}_{k|k-1}^s) \int dx_{k-1} \prod_{s=1}^S N(x_{k-1}; y^s, Y), \tag{28}$$

where

$$\begin{aligned}
x_{k|k-1}^s &= F_{k|k-1} \tilde{x}_{k-1|k-1}^s \\
\tilde{P}_{k|k-1}^s &= F_{k|k-1} \tilde{P}_{k-1|k-1} F_{k|k-1}^T + SQ_k \\
y^s &= Y \left(\left(\tilde{P}_{k|k-1}^s \right)^{-1} x_{k|k-1}^s + F_{k|k-1}^T (SQ_k)^{-1} x_k \right) \\
Y &= \left(\left(\tilde{P}_{k|k-1}^s \right)^{-1} + F_{k|k-1}^T (SQ_k)^{-1} F_{k|k-1} \right)^{-1}.
\end{aligned} \tag{29}$$

At this point, we have derived factorized prediction formulas for the DKF prediction, and to our knowledge, the remaining integral is part of the normalization constant. This, however, is

not trivial, since the parameter y^s depends on x_k . A successive application of the product formula yields the desired result:

$$\begin{aligned} \int dx_{k-1} \prod_{s=1}^S N\left(x_{k-1} - YF_{k|k-1}^T(SQ_k)^{-1}x_k; Y\left(\tilde{P}_{k|k-1}^s\right)^{-1}x_{k|k-1}^s, Y\right) \\ \propto \int dx_{k-1} N\left(x_{k-1} - YF_{k|k-1}^T(SQ_k)^{-1}x_k; \bar{\bar{y}}, \bar{\bar{Y}}\right) = 1, \end{aligned} \quad (30)$$

for some auxiliary variables $\bar{\bar{y}}$ and $\bar{\bar{Y}}$. All factors, which are independent of x_k have been omitted. This proves that the integral is independent of the state variable x_k . Concluding the derivations from above, we have derived the prediction formulas of the local estimation parameters as:

$$\begin{aligned} \tilde{x}_{k-1|k-1}^s &= SP_{k-1|k-1} \left(P_{k-1|k-1}^s\right)^{-1} x_{k-1|k-1}^s \\ \tilde{P}_{k-1|k-1} &= SP_{k-1|k-1} \\ x_{k|k-1}^s &= F_{k|k-1} \tilde{x}_{k-1|k-1}^s \\ \tilde{P}_{k|k-1}^s &= F_{k|k-1} \tilde{P}_{k-1|k-1} F_{k|k-1}^T + SQ_k. \end{aligned}$$

7.2. DKF filtering

Let Z_k denotes the set of measurements produced by all sensors at time t_k . The posterior density can be inferred by using the Bayes theorem:

$$p(x_k|Z^k) = \frac{p(Z_k|x_k)p(x_k|Z^{k-1})}{p(Z_k|Z^{k-1})} \quad (31)$$

Due to the mutual independence of the measurement noises, the joint likelihood function is given by:

$$p(Z_k|x_k) = \prod_{s=1}^S p(Z_k^s|x_k) \quad (32)$$

This is particularly useful for the structure of the product representation used for the DKF. Filling in the linear Gaussian models and neglecting the normalization constant in the denominator directly yields:

$$p(x_k|Z^k) \propto \prod_{s=1}^S N(Z_k; H_k^s x_k, R_k^s) N\left(x_k; x_{k|k-1}^s, \tilde{P}_{k|k-1}^s\right) \quad (33)$$

Thus, the product formula again can be applied to compute the posterior parameters:

$$p(x_k|Z^k) \propto \prod_{s=1}^S N\left(x_k; x_{k|k}^s, P_{k|k}^s\right) \quad (34)$$

where

$$\begin{aligned}
 x_{k|k}^s &= x_{k|k-1}^s + W_{k|k-1}^s v_k^s \\
 P_{k|k}^s &= P_{k|k-1}^s - W_{k|k-1}^s S_k^s W_{k|k-1}^{sT} \\
 v_k^s &= z_k^s - H_k^s x_{k|k-1}^s \\
 W_{k|k-1}^s &= P_{k|k-1}^s H_k^{sT} (S_k^s)^{-1} \\
 S_k^s &= H_k^s P_{k|k-1}^s H_k^{sT} + R_k^s.
 \end{aligned} \tag{35}$$

Again, we have omitted the factors, which are independent of x_k .

8. Information filter formulation of the DKF

In [6], an elegant derivation of the DKF formulas was published based on the *information filter* (IF). The IF uses information matrices, which are inverted covariances, and information states, which are information matrices multiplied with states. The optimal, centralized update formulas for S sensors based on the predicted information parameters $(P_{k|k-1})^{-1}$ and $(P_{k|k-1})^{-1} x_{k|k-1}$ are given by:

$$\begin{aligned}
 (P_{k|k})^{-1} x_{k|k} &= (P_{k|k-1})^{-1} x_{k|k-1} + \sum_{s=1}^S i_k^s \\
 (P_{k|k})^{-1} &= (P_{k|k-1})^{-1} + \sum_{s=1}^S I_k^s
 \end{aligned} \tag{36}$$

where $i_k^s = H_k^{sT} (R_k^s)^{-1} z_k^s$ and $I_k^s = H_k^{sT} (R_k^s)^{-1} H_k^s$ are the local information contribution from the current measurements at time t_k , which were received by the FC. If we want to distribute the computation to S nodes, we will have them uncorrelated as in the DKF previously. Since the fused estimate is obtained via the convex combination, the local information parameters are summed up:

$$\begin{aligned}
 (P_{k|k})^{-1} x_{k|k} &= \sum_{s=1}^S (P_{k|k}^s)^{-1} x_{k|k}^s \\
 (P_{k|k})^{-1} &= \sum_{s=1}^S (P_{k|k}^s)^{-1}
 \end{aligned} \tag{37}$$

This summation structure can be used to provide a closed prediction-filtering cycle.

8.1. Information DKF prediction

The prediction of the state is easier than a direct transition of the information parameters. Based on the fused estimate, one can obtain.

$$\begin{aligned}
 x_{k|k-1} &= F_{k|k-1} x_{k-1|k-1} \\
 &= F_{k|k-1} P_{k-1|k-1} \sum_{s=1}^S \left(P_{k-1|k-1}^s \right)^{-1} x_{k-1|k-1}^s \\
 &= \sum_{s=1}^S F_{k|k-1} P_{k-1|k-1} \left(P_{k-1|k-1}^s \right)^{-1} x_{k-1|k-1}^s.
 \end{aligned} \tag{38}$$

Thus, we have given the local predicted state parameters as:

$$x_{k|k-1}^s = F_{k|k-1} P_{k-1|k-1} \left(P_{k|k-1}^s \right)^{-1} x_{k|k-1}^s \tag{39}$$

Analogously, one obtains for the prior covariance:

$$\begin{aligned}
 P_{k|k-1} &= F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T + Q_k \\
 &= \frac{1}{S} F_{k|k-1} S P_{k-1|k-1} F_{k|k-1}^T + S Q_k
 \end{aligned} \tag{40}$$

Thus, if we set $P_{k|k-1}^s = S P_{k|k-1} = F_{k|k-1} S P_{k-1|k-1} F_{k|k-1}^T + S Q_k$, then the convex combination yields the exact global fused covariance.

8.2. Information DKF filtering

For the filtering, it is assumed that each sensor has computed its local information contribution parameter i_k^s and I_k^s from its own sensor model and, in addition, the information matrix contributions I_k^l from all remote sensors l by using the individual sensor models which are again assumed to be known. Then, the information state is updated via.

$$\begin{aligned}
 (P_{k|k})^{-1} x_{k|k} &= (P_{k|k-1})^{-1} x_{k|k-1} + \sum_{s=1}^S i_k^s \\
 &= \sum_{s=1}^S \left(P_{k|k-1}^s \right)^{-1} x_{k|k-1}^s + i_k^s.
 \end{aligned} \tag{41}$$

As a direct consequence, the updated parameters of the local processors follow the standard IF filtering equations:

$$\left(P_{k|k}^s \right)^{-1} x_{k|k}^s = \left(P_{k|k-1}^s \right)^{-1} x_{k|k-1}^s + i_k^s. \tag{42}$$

For the globalized information matrix, the remote information parameters from the sensor models are used:

$$\begin{aligned}
(P_{k|k})^{-1} &= (P_{k|k-1})^{-1} + \sum_{s=1}^S I_k^s \\
&= \sum_{s=1}^S (P_{k|k-1}^s)^{-1} + I_k^s \\
(P_{k|k}^s)^{-1} &= (P_{k|k-1}^s)^{-1} + I_k^s
\end{aligned} \tag{43}$$

It is important to note that the local processing nodes compute both the local pseudo information matrix $(P_{k|k}^s)^{-1}$ and the global fused error covariance $P_{k|k}$, which is required for the prediction to the next time step.

9. Distributed accumulated state density filter

The DKF from Sections 7 and 8 can be considered a big step toward distributed state estimation, tracking, and information inference. However, in practical applications, the exact solution is often hindered by the fact that the exact remote sensor model parameters are unknown and can only be approximated based on local state estimates. The good news is that there is another exact solution based on the *accumulated state density* (ASD). The distributed ASD equations turn the spatial correlations into temporal correlations of successive states. Originally, the ASD equations were introduced to solve the out-of-sequence problem, which handles delayed transmissions of measurements into an ongoing fusion process in an optimal manner. Therefore, the temporal correlations can well be coped with the ASD approach.

At first, let us introduce the ASD state $x_{k:n}$ as

$$x_{k:n} = (x_k^T, \dots, x_n^T)^T \tag{44}$$

where t_k refers to the current time of the filtering process and t_n refers to the initialization time. The ASD approach now considers the conditional joint density $p(x_{k:n}|Z^k)$, that is, the posterior of the full trajectory of the target between t_n and t_k . In particular, the individual state densities of a single instant of time can be obtained via marginalization. Also, it should be noted that the *Rauch-Tung-Striebel* (RTS) smoothing equations are inherently integrated in the ASD posterior, since all states are conditioned on the full set of measurements up to time t_k [7].

A recursive computation of the ASD posterior can be achieved by using the Bayes theorem:

$$p(x_{k:n}|Z^k) = \frac{p(Z_k|x_{k:n})p(x_{k:n}|Z^{k-1})}{p(Z_k|Z^{k-1})} \tag{45}$$

Since the measurements conditioned on the whole trajectory only depend on the state at time t_k , the joint likelihood function is given by:

$$p(Z_k|x_{k:n}) = p(Z_k|x_k) = \prod_{s=1}^S p(z_k^s|x_k) \quad (46)$$

The second factor can be reformulated as follows:

$$\begin{aligned} p(x_{k:n}|Z^{k-1}) &= p(x_k|x_{k-1:n}, Z^{k-1})p(x_{k-1:n}|Z^{k-1}) \\ &= p(x_k|x_{k-1})p(x_{k-1:n}|Z^{k-1}) \end{aligned} \quad (47)$$

where we have used the Markov property of the system in the last equation. This recursive representation can now be repeated on the term $p(x_{k-1:n}|Z^{k-1})$. A successive application of this procedure yields

$$\begin{aligned} p(x_{k:n}|Z^k) &\propto \prod_{s=1}^S \{p(z_k^s|x_k)\} p(x_k|x_{k-1}) \prod_{s=1}^S \{p(z_{k-1}^s|x_{k-1})\} p(x_{k-1}|x_{k-2}) \cdots \\ &\quad \prod_{s=1}^S \{p(z_{n+1}^s|x_{n+1})\} p(x_{n+1}|x_n) \cdot p(x_n|Z^n) \end{aligned} \quad (48)$$

where we have neglected the normalization constant in the denominator. Filling in our Gaussian models and using the factorization of the transition model from above equation yields

$$p(x_{k:n}|Z^k) \propto \prod_{l=n+1}^k \prod_{s=1}^S \{N(z_l^s; H_l^s x_l, R_l^s) N(x_l; F_{l|l-1} x_{l-1}, S Q_l)\} \cdot p(x_n|Z^n) \quad (49)$$

Since the initial density usually is based on a first measurement, we can assume that it factorizes into independent local track starts:

$$p(x_n|Z^n) \propto \prod_{s=1}^S N(x_n; x_{n|n}^s, P_{n|n}^s) \quad (50)$$

When the posterior is fully factorized in the number of sensors and in the time steps, each processing node can compute the resulting ASD Gaussian with mean $x_{k:n|k}^s$ and covariance matrix $P_{k:n|k}^s$ [7]:

$$p(x_{k:n}|Z^k) \propto \prod_{s=1}^S N(x_{k:n}; x_{k:n|k}^s, P_{k:n|k}^s) \quad (51)$$

where the parameters are given by:

$$x_{k:n|k}^s = \left(x_{k|k}^{s\top}, \dots, x_{n|k}^{s\top} \right)^\top,$$

$$P_{k:n|k}^s = \begin{pmatrix} P_{k|k}^s & P_{k|k}^s W_{k-1|k}^{s\top} & P_{k|k}^s W_{k-2|k}^{s\top} & \dots & P_{k|k}^s W_{n|k}^{s\top} \\ W_{k-1|k}^s P_{k|k}^s & P_{k-1|k}^s & P_{k-1|k}^s W_{k-2|k-1}^{s\top} & * & P_{k-1|k}^s W_{n|k-1}^{s\top} \\ W_{k-2|k}^s P_{k|k}^s & W_{k-2|k-1}^s P_{k-1|k}^s & P_{k-2|k}^s & * & P_{k-2|k}^s W_{n|k-2}^{s\top} \\ \vdots & * & * & * & \vdots \\ W_{n|k}^s P_{k|k}^s & W_{n|k-1}^s P_{k-1|k}^s & \dots & W_{n|+1}^s P_{n+1|k}^s & P_{n|k}^s \end{pmatrix}$$

We have used a short notation such that $x_{l|k} = E[x_l|z^k]$ are the smoothed state estimates and $P_{l|k} = \text{cov}[x_l|z^k]$ are the covariances, respectively, which result from the Rauch-Tung-Striebel equations. Also, the combined retrodiction gain matrices are known from the RTS smoother:

$$W_{l|k} = \prod_{i=l}^{k-1} W_{i|i+1} \quad (52)$$

$$W_{i|i+1} = P_{i|i} F_{i+1|i}^\top (P_{i+1|i})^{-1}$$

Thus, when the FC receives the local ASD parameters, the optimal fused estimate can be obtained via the convex combination:

$$x_{k:n|k} = P_{k:n|k} \sum_{s=1}^S \left(P_{k:n|k}^s \right)^{-1} x_{k:n|k}^s, \quad (53)$$

$$P_{k:n|k} = \left(\sum_{s=1}^S \left(P_{k:n|k}^s \right)^{-1} \right)^{-1}$$

For a continuous state estimation process, it is convenient to formulate the distributed ASD solution in terms of a prediction-filtering cycle.

9.1. Distributed ASD prediction

For the prediction step, it is assumed that the local processing node s has computed the previous filtering parameters $x_{k-1:n|k-1}^s$ and $P_{k-1:n|k-1}^s$, which refer to time t_{k-1} . Then, the prior parameters are given by:

$$x_{k:n|k-1}^s = \left(x_{k|k-1}^{s\top}, x_{k-1:n|k-1}^{s\top} \right)^\top$$

$$P_{k:n|k-1}^s = \begin{pmatrix} P_{k|k-1}^s & P_{k-1:n|k-1}^s W_{k-1:n}^{s\top} \\ W_{k-1:n}^s P_{k-1:n|k-1}^s & P_{k-1:n|k-1}^s \end{pmatrix} \quad (54)$$

$$x_{k|k-1}^s = F_{k|k-1} x_{k-1|k-1}^s$$

$$P_{k|k-1}^s = F_{k|k-1} P_{k-1|k-1}^s F_{k|k-1}^T + S Q_k$$

$$W_{k-1:n}^s = \begin{pmatrix} W_{k-1|k}^s \\ W_{k-2|k}^s \\ \vdots \\ W_{n|k}^s \end{pmatrix}$$

9.2. Distributed ASD filtering

Since the prior is factorized in form of a product representation and the current measurements from time t_k are mutually uncorrelated, local Kalman filters can be applied to obtain the posterior parameters:

$$x_{k:n|k}^s = x_{k:n|k-1}^s + W_{k:n|k}^s \left(z_k^s - H_k^s \Pi_k x_{k:n|k-1}^s \right)$$

$$P_{k:n|k}^s = P_{k:n|k-1}^s - W_{k:n|k}^s S_k^s W_{k:n|k}^{sT}$$

$$S_k^s = H_k^s \Pi_k P_{k:n|k-1}^s \Pi_k^T H_k^{sT} + R_k^s$$

$$W_{k:n|k}^s = P_{k:n|k-1}^s \Pi_k^T H_k^{sT} (S_k^s)^{-1} \quad (55)$$

10. Conclusion

In this chapter, we have introduced the least squares solution to the track-to-track fusion problem, where cross-covariances of the track estimation errors are required. Neglecting the cross-covariances has led us to the Naïve fusion, a simple but powerful fusion algorithm for practical applications. By recursive computation of the cross-covariances, we have seen that they primarily depend on the process noise of the state transition kernel. Since a centralized computation of the cross-covariances is infeasible in practical applications, more sophisticated solutions are required for optimal fusion results. The distributed Kalman filter, which uses the product representation to keep the local parameters decorrelated achieved this. However, this approach only works, if the local processors know all measurement models at each time step. Then, the distributed accumulated state density filter uses the temporal correlations to factorize the global posterior density. This approach does not require remote sensor models and is therefore, well suited for extensions with measurement ambiguity or nonlinear measurement functions.

In the study, one can find more extensions based on the distributed Kalman filter to overcome the lack of knowledge on the remote sensor models. In [8] and the references therein, a debiasing matrix is introduced to compensate for globally biased gain matrices of the local filters. An application of the tracklet fusion based on the distributed accumulated state density filter can be found in [9]. Then, in [10], the information filter formulation of the distributed Kalman filter also was extended to scenarios with input information on the transition process.

A. Appendix

A.1. Matrix inversion lemma

Let A, B, C , and D be matrices of a block matrix such that A and D are invertable, and also such that the Schur-complements $D - CA^{-1}B$ and $A - BD^{-1}C$ have full rank. Then, the inversion of the block matrix is given by:

$$\begin{aligned} \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} &= \begin{pmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{pmatrix} \\ &= \begin{pmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{pmatrix}. \end{aligned} \quad (56)$$

In particular, it holds that

$$\begin{aligned} (A - BD^{-1}C)^{-1} &= A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} - (A - BD^{-1}C)^{-1}BD^{-1} = \\ &= -A^{-1}B(D - CA^{-1}B)^{-1} - D^{-1}C(A - BD^{-1}C)^{-1} = -(D - CA^{-1}B)^{-1}CA^{-1}D^{-1} \\ &+ D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} = (D - CA^{-1}B)^{-1} \end{aligned} \quad (57)$$

Proof. Let the inverted block matrix be given by submatrices E, F, G , and H . By definition, it holds that

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} I & O \\ O & I \end{pmatrix} \text{ and } \begin{pmatrix} E & F \\ G & H \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & O \\ O & I \end{pmatrix},$$

where I and O are the identity and zero matrix, respectively. A matrix multiplication of the first and second equality yields two blocks of equations:

$$\begin{aligned} AE + BG &= I \\ AF + BH &= O \\ CE + DG &= O \\ CF + DH &= I \end{aligned} \quad (58)$$

which we call *block A* and

$$\begin{aligned} EA + FC &= I \\ EB + FD &= O \\ GA + HC &= O \\ GB + HD &= I \end{aligned} \quad (59)$$

which we call *block B*. Resolving block A for E, F, G , and H yields the first version of the inverted block matrix, whereas resolving block B yields the second version.

A.2. Product formula for Gaussian densities

For two Gaussian distributed random variables x and z , it holds that

$$N(x; y, P)N(z; Hx, R) = N(x; \bar{y}, \bar{P})N(z; \bar{z}, S) \quad (60)$$

where

$$\begin{aligned} \bar{y} &= \begin{cases} y + Wv \\ P^{-1}y + H^T R^{-1}z \end{cases} \\ \bar{P} &= \begin{cases} P - WSW^T \\ (P^{-1} + H^T R^{-1}H)^{-1} \end{cases} \\ S &= HPH^T + R \\ W &= PH^T S^{-1} \\ v &= z - Hy \end{aligned} \quad (61)$$

A proof can be found, for instance, in [11].

Author details

Felix Govaers

Address all correspondence to: felix.govaers@fkie.fraunhofer.de

Fraunhofer FKIE, Wachtberg, Germany

References

- [1] Chong CY. Hierarchical estimation. Proceedings of the 2nd MIT/ONR Workshop on Distributed Information and Decision Systems Motivated by Naval Command Control Communication Systems, Monterey, CA, USA. 1979
- [2] Mahmoud MS, Khalid HM. Distributed Kalman Filtering: A Bibliographic Review. IET Control Theory & Applications; 2013
- [3] Campbell ME, Ahmed NR. Distributed Data Fusion: Neighbors, Rumors, and the Art of Collective Knowledge. IEEE Control Systems; 2016
- [4] Hall D, Chong C-Y, Llinas J, Martin Liggins II, editors. Distributed Data Fusion for Network-Centric Operations. CRC Press; 2013
- [5] Govaers F, Koch W. An exact solution to track-to-track-fusion at arbitrary communication rates. In: IEEE Transactions on Aerospace and Electronic Systems. Vol. 48. 2012

- [6] Chong CY, Mori S. Optimal Fusion For Non-Zero Process Noise. Proceedings of the 16th International Conference on Information Fusion, Istanbul. 2013
- [7] Koch W, Govaers F. On accumulated state densities with applications to out-of-sequence measurement processing. In: IEEE Transactions on Aerospace and Electronic Systems. Vol. 47. 2011
- [8] Reinhardt M. Linear Estimation in Interconnected Sensor Systems with Information Constraints. PhD thesis: KIT Scientific Publishing; 2014
- [9] Govaers F, Chong CY, Mori S, Koch W. Comparison of Augmented State Track Fusion Methods for Non-Full-Rate Communication. 18th International Conference on Information Fusion, Washington, DC. 2015
- [10] Pfaff F, Noack B, Hanebeck UD, Govaers F, Koch W. Information Form Distributed Kalman Filtering (IDKF) with Explicit Inputs. 20th International Conference on Information Fusion, Xi'an, China. 2017
- [11] Koch W. Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications. Springer; 2014:251