

## An overview of non-centralized Kalman filters

**Citation for published version (APA):**

Sijs, J., Lazar, M., Bosch, van den, P. P. J., & Papp, Z. (2008). An overview of non-centralized Kalman filters. In *Control Applications, 2008. CCA 2008. IEEE International Conference, San Antonio, Texas, USA, September 3-5, 2008* (pp. 738-744). Institute of Electrical and Electronics Engineers.  
<https://doi.org/10.1109/CCA.2008.4629588>

**DOI:**

[10.1109/CCA.2008.4629588](https://doi.org/10.1109/CCA.2008.4629588)

**Document status and date:**

Published: 01/01/2008

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# An overview of non-centralized Kalman filters

J. Sijs, *Student Member, IEEE*

M. Lazar

P.P.J. van den Bosch

Z. Papp

**Abstract**—The usage of Wireless Sensor Networks (WSNs) for state-estimation has recently gained increasing attention due to its cost effectiveness and feasibility. One of the major challenges of state-estimation via WSNs is the distribution of the centralized state-estimator among the nodes in the network. Significant emphasis has been on developing non-centralized state-estimators considering communication, processing-demand and estimation-error. This survey paper presents different methodologies to obtain non-centralized state-estimators and focuses on the estimation algorithms and their implementation. The temperature distribution of a bar is used as a benchmark to assess the non-centralized state-estimators in terms of estimation-error and communication requirements.

**Index Terms**—Wireless Sensor Networks, distributed state-estimation, Kalman filter.

## I. INTRODUCTION

State-estimation is a widely used technique in monitoring and control applications. An important state-estimator still widely used today is the Kalman filter formally presented in [1]. The method requires that all process-measurements are sent to a central system which estimates the global state-vector of the process. The interest for using WSNs to retrieve the measurements has recently grown [2], due to improved performance and feasibility in new application areas. However, for WSNs consisting of a large amount of nodes a central state-estimator becomes impracticable due to high processing demand and energy consumption. As a result, the distribution of the centralized Kalman filter, in which each node estimates its own state-vector, has become a challenging and active research area.

Within this research area two different directions can be noticed. In one direction each node estimates the global state-vector and a central system is used to fuse the information of all the nodes together. Examples of such methods, also called sensor fusion (SF) can be found in [3]–[8]. In the second direction the central estimation is absent. Instead each node estimates a part of the global state-vector using information from other nodes in its local region, preferably its direct neighbors. Articles that describe these distributed Kalman filters (DKFs) are [9]–[14].

The purpose of this paper is to provide a critical overview of existing non-centralized Kalman filters, which would help in choosing a particular method for a particular application.

J. Sijs and Z. Papp are with TNO Science and Industry, P.O. Box 155, 2600 AD Delft, The Netherlands, E-mail: joris.sijs@tno.nl, zoltan.papp@tno.nl.

M. Lazar and P.P.J. van den Bosch are with the Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, E-mail: m.lazar@tue.nl, p.p.j.v.d.bosch@tue.nl.

For each method we present its characteristics, algorithm and amount of decentralization in terms of processing demand and communication requirements per node. Finally all methods are assessed in a benchmark problem on their performance in estimation, communication and robustness to data loss or node break down.

The remainder of the paper has the following structure. Some basic notation and the principles of the centralized Kalman filter are described in Section II. The initial SF/DKF presented in [3] is then described in Section III. This method was later used to design DKFs, as shown in Section IV. Section V presents a hierarchical Kalman filter, while a DKF with weighted averaging is given in Section VI. Section VII discusses a DKF with consensus filters and a DKF with bipartite fusion graphs is presented in Section VIII. Finally, the different non-centralized Kalman filters are assessed in Section IX using the benchmark example of estimating the temperature of a bar via a wireless sensor network. Conclusions and recommendations are formulated in Section X.

## II. CENTRALIZED KALMAN FILTER

Suppose a WSN is used in combination with a centralized Kalman filter to estimate the states of a global process. All nodes send their measurements to one system where the centralized Kalman filter estimates the global state-vector. The measurements of the  $k^{th}$  sample instant are combined in the measurement-vector  $y[k]$  with measurement-noise  $v[k]$ . The global state-vector of the process is defined as  $x[k]$  with process-noise  $w[k]$ . With this, the discretised process-model becomes:

$$x[k] = Ax[k-1] + w[k-1], \quad (1a)$$

$$y[k] = Cx[k] + v[k]. \quad (1b)$$

The probability density function (PDF) of both  $w[k]$  and  $v[k]$  are described by a Gaussian-distribution, i.e.

$$\begin{aligned} E(w[k]) &= 0 \text{ and } E(w[k]w^T[k]) = Q[k], \\ E(v[k]) &= 0 \text{ and } E(v[k]v^T[k]) = R[k]. \end{aligned} \quad (2)$$

The centralized Kalman filter [1] estimates the global state-vector  $\hat{x}[k]$  and the global error-covariance matrix  $P[k]$ . Let  $E(\alpha)$  represent the expectation of the stochastic variable  $\alpha$ . Then,  $\hat{x}[k]$  and  $P[k]$  are defined as:

$$\hat{x}[k] = E(x[k]), \quad P[k] = E((x[k] - \hat{x}[k])(x[k] - \hat{x}[k])^T). \quad (3)$$

The centralized Kalman filter consists of two stages that are performed each sample instant  $k$ : the “prediction-step” and the “measurement-update”. First the prediction-step computes the predicted state-vector  $\hat{x}[k|k-1]$  and error-

covariance  $P[k|k-1]$ . Second, the measurement-update calculates the estimated state-vector  $\hat{x}[k|k]$  and error-covariance  $P[k|k]$ . The centralized Kalman filter, with initial values  $\hat{x}[0|0] = x_0$  and  $P[0|0] = P_0$ , is formally described by the following set of equations:

$$\begin{aligned} &\text{prediction-step} \\ &\hat{x}[k|k-1] = A\hat{x}[k-1|k-1], \\ &P[k|k-1] = AP[k-1|k-1]A^T + Q[k-1], \end{aligned} \quad (4a)$$

$$\begin{aligned} &\text{measurement-update} \\ &K[k] = P[k|k-1]C^T(CP[k|k-1]C^T + R[k])^{-1}, \\ &\hat{x}[k|k] = \hat{x}[k|k-1] + K[k](y[k] - C\hat{x}[k|k-1]), \\ &P[k|k] = (I - K[k]C)P[k|k-1]. \end{aligned} \quad (4b)$$

For large scale WSNs the centralized implementation of (4) results in high processing demand, communication requirements and energy consumption, which prevents the usage of a centralized Kalman filter. To overcome this issue, a number of methodologies to implement the Kalman filter in a distributed fashion were designed. However, until now there has been no comparison or evaluation of the obtained results in this direction. The purpose of this paper is to provide a critical overview of existing methods for designing non-centralized Kalman filters. The performance of the different DKFs is illustrated using a benchmark application example in Section IX.

Before explaining the different methods of this overview in detail, we present three assumptions. If not indicated otherwise, these assumptions hold for the presented method. Firstly, the existence of a WSN consisting of  $N$  nodes is assumed in which each node  $i$  has its own measurement-vector  $y_i$  with corresponding measurement-noise  $v_i$ . The global measurement-vector  $y$ , observation-matrix  $C$  and equation (1b) are rewritten as follows:

$$y_i[k] = C_i x[k] + v_i[k] \Rightarrow \begin{cases} y = (y_1, y_2, \dots, y_N)^T \\ C = (C_1, C_2, \dots, C_N)^T. \end{cases} \quad (5)$$

Secondly, the measurement-noises of two different nodes are uncorrelated, i.e.  $R_{(i,j)} = E(v_i v_j^T) = 0$ , if  $i \neq j$ . Resulting in an  $R$ -matrix of the form:

$$R = \text{blockdiag}(R_{(1,1)}, R_{(2,2)}, \dots, R_{(N,N)}). \quad (6)$$

Thirdly, all nodes  $j$  that are directly connected to a node  $i$  are collected in the set  $N_i$ , which also includes the node  $i$ . This means that if node  $j$  is connected to node  $i$ , then  $j \in N_i$ . Usually,  $N_i$  is containing only direct neighbors of node  $i$ . However, it is also possible that  $N_i$  contains other nodes besides the direct neighbors and in the case of global communication  $N_i = N$ . This will be made clear for each estimation method.

### III. PARALLEL INFORMATION FILTER

This section describes a parallel implementation of the Kalman filter [3]. Each node  $i$  has its own Kalman filter calculating the global state-estimates  $\hat{x}_i$  and  $P_i$  of node  $i$  using only its measurement-vector  $y_i$ . In the algorithm

an information-matrix  $I_i$  and an information-vector  $i_i$  are computed from the  $y_i$  and  $R_{(i,i)}$ . Each node sends its state-estimates to a central system which calculates the global state-estimates of the whole WSN, i.e.  $\hat{x}$  and  $P$ .

The sets of equations of the parallel information filter(PIF) for node  $i$  are:

$$\begin{aligned} &\text{node } i \text{ prediction-step} \\ &\hat{x}_i[k|k-1] = A\hat{x}_i[k-1|k-1], \\ &P_i[k|k-1] = AP_i[k-1|k-1]A^T + Q[k-1], \end{aligned} \quad (7a)$$

$$\begin{aligned} &\text{node } i \text{ information-update} \\ &I_i[k] = C_i^T R_{(i,i)}^{-1}[k]C_i, \quad i_i[k] = C_i^T R_{(i,i)}^{-1}[k]y_i[k], \end{aligned} \quad (7b)$$

$$\begin{aligned} &\text{node } i \text{ measurement-update} \\ &P_i^{-1}[k|k] = P_i^{-1}[k|k-1] + I_i[k], \\ &\hat{x}_i[k|k] = P_i[k|k](P_i^{-1}[k|k-1]\hat{x}_i[k|k-1] + i_i[k]). \end{aligned} \quad (7c)$$

The global state-estimates  $\hat{x}$  and  $P$  are calculated taking the covariance intersection into account [15]:

$$\begin{aligned} \alpha_i[k] &= \frac{(tr(P_i[k|k]))^{-1}}{\sum_{i=1}^N (tr(P_i[k|k]))^{-1}}, \quad P^{-1}[k] = \sum_{i=1}^N \alpha_i[k]P_i^{-1}[k|k], \\ \hat{x}[k] &= \sum_{i=1}^N \alpha_i[k]P_i[k|k]\hat{x}_i[k|k]. \end{aligned} \quad (8)$$

The calculation of  $\hat{x}[k]$  and  $P[k]$  is done in a central system, which can be located in one node only or even in every node. A drawback of this method is that every node estimates a global state-vector leading to a high processing-demand. A second drawback is global communication, for every node needs to send information to at least one central system. This method was improved in the decentralized information filter presented in the next section.

### IV. DECENTRALIZED INFORMATION FILTER

In [9] the decentralized information filter (DIF) was proposed to overcome some drawbacks of the PIF. Again each node  $i$  has its own global state-estimates  $\hat{x}_i$  and  $P_i$ . However, the central estimation is decentralized among the nodes and a node  $i$  is only connected to its neighboring nodes in  $N_i$ . These nodes exchange their information-matrix  $I_i$  and information-vector  $i_i$ . Meaning that node  $i$  receives  $I_j$  and  $i_j$  from the nodes  $j$  with  $j \in N_i$ ,  $j \neq i$ . The received  $I_j$  and  $i_j$  are added to  $I_i$  and  $i_i$  respectively.

The sets of equations of the DIF for node  $i$  are:

$$\begin{aligned} &\text{node } i \text{ prediction-step} \\ &\hat{x}_i[k|k-1] = A\hat{x}_i[k-1|k-1], \\ &P_i[k|k-1] = AP_i[k-1|k-1]A^T + Q[k-1], \end{aligned} \quad (9a)$$

$$\begin{aligned} &\text{node } i \text{ information-update} \\ &I_i[k] = C_i^T R_{(i,i)}^{-1}[k]C_i, \quad i_i[k] = C_i^T R_{(i,i)}^{-1}[k]y_i[k], \end{aligned} \quad (9b)$$

$$\begin{aligned} &\text{local measurement-update} \\ &P_i^{-1}[k|k] = P_i^{-1}[k|k-1] + \sum_{j \in N_i} I_j[k], \\ &\hat{x}_i[k|k] = P_i[k|k](P_i^{-1}[k|k-1]\hat{x}_i[k|k-1] + \sum_{j \in N_i} i_j[k]). \end{aligned} \quad (9c)$$

An important aspect of this DKF is that if node  $i$  is connected to all other nodes and assumptions (5) and (6) are valid, its state-estimates  $\hat{x}_i$  and  $P_i$  are exactly the same as the estimates of a centralized Kalman filter [1]. An advantage is that only local communication is required. A drawback however, is that each node estimates the global state-vector.

#### V. DECOUPLED HIERARCHICAL KALMAN FILTER

In [10]–[12] decoupled hierarchical Kalman filters (DHKFs) are presented. The common feature of this method is that the global state-vector  $x$  and the process-model are divided in  $N$  parts. Each node estimates one of the  $N$  parts and exchanges its state-estimates with all other nodes in the WSN. The process-model is described as:

$$\begin{pmatrix} x_1[k] \\ \vdots \\ x_N[k] \end{pmatrix} = A \begin{pmatrix} x_1[k-1] \\ \vdots \\ x_N[k-1] \end{pmatrix} + \begin{pmatrix} w_1[k-1] \\ \vdots \\ w_N[k-1] \end{pmatrix}, \quad (10)$$

$$\begin{pmatrix} y_1[k] \\ \vdots \\ y_N[k] \end{pmatrix} = C \begin{pmatrix} x_1[k] \\ \vdots \\ x_N[k] \end{pmatrix} + \begin{pmatrix} v_1[k] \\ \vdots \\ v_N[k] \end{pmatrix},$$

where,

$$A = \begin{pmatrix} A_{(1,1)} & \cdots & A_{(1,N)} \\ \vdots & \ddots & \vdots \\ A_{(N,1)} & \cdots & A_{(N,N)} \end{pmatrix}, C = \begin{pmatrix} C_{(1,1)} & \cdots & C_{(1,N)} \\ \vdots & \ddots & \vdots \\ C_{(N,1)} & \cdots & C_{(N,N)} \end{pmatrix}.$$

Just as  $R$ , also the matrices  $Q$  and  $P$  are both assumed to be block-diagonal matrices. Therefore we define  $Q_i = E(w_i w_i^T)$  and  $P_i = E((x_i - \hat{x}_i)(x_i - \hat{x}_i)^T)$ . Node  $i$  estimates  $\hat{x}_i[k]$  and  $P_i[k]$ . The algorithm for each node  $i$  is:

node  $i$  prediction-step

$$\hat{x}_i[k|k-1] = \sum_{j=1}^N A_{(i,j)} \hat{x}_j[k-1|k-1], \quad (11a)$$

$$P_i[k|k-1] = \sum_{j=1}^N (A_{(i,j)} P_j[k-1|k-1] A_{(i,j)}^T) + Q_i[k-1],$$

node  $i$  measurement-update

$$K_i[k] = P_i[k|k-1] C_{(i,i)}^T \left( \sum_{j=1}^N (C_{(i,j)} P_j[k|k-1] C_{(i,j)}^T) + R_{(i,i)}[k] \right)^{-1},$$

$$\hat{x}_i[k|k] = \hat{x}_i[k|k-1] + K_i[k] (y_i[k] - \sum_{j=1}^N C_{(i,j)} \hat{x}_j[k|k-1]),$$

$$P_i[k|k] = (I - K_i[k] C_{(i,i)}) P_i[k|k-1]. \quad (11b)$$

Notice that this method is better compared to the PIF and DIF in terms of processing-demand and the amount of data transfer required. A drawback however, is that global communication is still required.

#### VI. DISTRIBUTED KALMAN FILTER WITH WEIGHTED AVERAGING

In previous methods each node sends a vector with its corresponding covariance-matrix to the other nodes, i.e.  $i$

with  $I_i$  or  $\hat{x}_i$  with  $P_i$ . In the distributed Kalman filter with weighted averaging (DKF-WA) [13] a node  $i$  only sends its state-vector, without covariance-matrix, to its neighboring nodes in the set  $N_i$ . The weighted average of all received state-vectors forms the node's estimated global state-vector  $\hat{x}_i$ . One remark should be made: in this case the matrix  $R$  is not necessarily block-diagonal, i.e.  $R_{(i,j)} \neq 0, \forall j \in N_i$ .

The algorithm of the DKF-WA is divided into an on-line and an off-line part. In the on-line part each node has its own estimate of the global state-vector  $\hat{x}_i$  which is partly calculated using the equations of the centralized Kalman filter. In this method a node  $i$  has a fixed, pre-calculated Kalman gain  $K_i$ . After the measurement-update the nodes exchange their estimated state-vector. A node  $i$  receives the state-vectors  $\hat{x}_j$  ( $j \in N_i$ ) which are then weighted with a fixed, pre-calculated matrix  $W_{(i,j)}$ . The weighted average is chosen as the new estimated global state-vector of node  $i$ , i.e.  $\hat{x}_i$ . The on-line algorithm is:

node  $i$  prediction-step (on-line)

$$\hat{x}_i[k|k-1] = A \hat{x}_i[k-1|k-1], \quad (12a)$$

node  $i$  measurement-update (on-line)

$$\hat{x}_i[k|k] = \hat{x}_i[k|k-1] + K_i (y_i[k] - C_i \hat{x}_i[k|k-1]), \quad (12b)$$

local weighted average (on-line)

$$\hat{x}_i[k|k] = \sum_{j \in N_i} W_{(i,j)} \hat{x}_j[k|k]. \quad (12c)$$

Next, we explain the off-line algorithm which is used to calculate  $K_i$  and  $W_{(i,j)}$ . For that, the error-covariance between the estimated global state-vectors of node  $i$  and  $j$  is:

$$P_{(i,j)}[k] = E((x[k] - \hat{x}_i[k])(x[k] - \hat{x}_j[k])^T). \quad (13)$$

The off-line algorithm uses the same stages for  $P_{(i,j)}$  as the on-line algorithm for  $\hat{x}_i$  in (12). First the ‘‘prediction-step’’ (12a) and ‘‘measurement-update’’ (12b) of a node  $i$  are given to calculate  $P_{(i,j)}[k|k-1]$  and  $P_{(i,j)}[k|k]$ , with  $j \in N_i$ :

node  $i$  prediction-step (off-line)

$$P_{(i,j)}[k|k-1] = A P_{(i,j)}[k-1|k-1] A^T + Q[k-1], \quad (14a)$$

node  $i$  measurement-update (off-line)

$$\begin{aligned} K_i[k] &= P_{(i,i)}[k|k-1] C_i^T (C_i P_{(i,i)}[k|k-1] C_i^T + R_{(i,i)}[k])^{-1}, \\ P_{(i,j)}[k|k] &= (I - K_i[k] C_i) P_{(i,j)}[k|k-1] (I - K_j[k] C_j)^T \\ &\quad + K_i[k] R_{(i,j)} K_j^T[k]. \end{aligned} \quad (14b)$$

Notice  $R_{(i,j)}$  and the calculation of  $K_i$  in (14b). The next step is calculating  $W_{(i,j)}$  of the weighted average as in (12c). To keep the state-estimation unbiased the following constraint is introduced:

$$\sum_{j \in N_i} W_{(i,j)}[k] = I_{n \times n}. \quad (15)$$

From (12c) and (15) we can derive:

$$\begin{aligned} x[k] - \hat{x}_i[k|k] &= \sum_{j \in N_i} W_{(i,j)} x[k] - \sum_{j \in N_i} W_{(i,j)} \hat{x}_j[k|k], \\ &= \sum_{j \in N_i} W_{(i,j)} (x[k] - \hat{x}_j[k|k]). \end{aligned} \quad (16)$$

Using (13) the weighted average of  $P_{(i,j)}[k|k]$  results in:

$$P_{(i,j)}[k|k] = \sum_{p \in N_i} \sum_{q \in N_j} W_{(i,p)}[k] P_{(p,q)}[k|k] W_{(j,q)}^T[k]. \quad (17)$$

Equation (17) can also be written in matrix form. If  $N_i = (i_1, i_2, \dots, i_{N_i})$  and we define  $\bar{W}_i = (W_{(i,i_1)}, \dots, W_{(i,i_{N_i})})$ , equation (17) becomes:

$$P_{(i,j)}[k|k] = \bar{W}_i \begin{pmatrix} P_{(i_1,j_1)}[k|k] & \cdots & P_{(i_1,j_{N_j})}[k|k] \\ \vdots & \ddots & \vdots \\ P_{(i_{N_i},j_1)}[k|k] & \cdots & P_{(i_{N_i},j_{N_j})}[k|k] \end{pmatrix} \bar{W}_j^T. \quad (18)$$

The last step in this off-line algorithm is to minimize  $P_{(i,j)}[k|k]$  with respect to  $\bar{W}_i = (W_{(i,i_1)}, \dots, W_{(i,i_{N_i})})$  taking constraint (15) into account. For further details we refer the interested reader to [13]. The off-line algorithm runs until the values  $K_i$  and  $W_{(i,j)}$  remain constant. These values are then used in the on-line algorithm.

An important aspect in the performance of this method is that each node estimates the global state-vector, but due to the fixed matrices  $K_i$  and  $W_{(i,j)}$  its processing-demand remains low. It was already noticed that the DKF-WA has low communication requirements. However, it is not robust against lost data or nodes breaking down. For in that case the weighted averaging of (12c) will not be accurate.

## VII. DISTRIBUTED KALMAN FILTER WITH CONSENSUS FILTERS

In [14], [16] the distributed Kalman filter with consensus filter (DKF-CF) was proposed. In this method a node  $i$  has its own estimate of the global state-vector  $\hat{x}_i$  and the node can only communicate with its neighboring nodes collected in  $N_i$ . Instead of averaging the received state-vectors, a node tries to reach consensus on them using a correction-factor  $\varepsilon$ .

Basically the algorithm of the DKF-CF adds an extra stage to the algorithm of the DIF in (9), i.e. the ‘‘local-consensus’’-stage. Hence, every node has its own global state-estimates  $\hat{x}_i$  and  $P_i$ . We define  $\hat{x}_i^c$  to be the estimated global state-vector of node  $i$  before the consensus-stage. The algorithm is:

$$\begin{aligned} &\text{node } i \text{ prediction-step} \\ \hat{x}_i[k|k-1] &= A\hat{x}_i[k-1|k-1], \end{aligned} \quad (19a)$$

$$\begin{aligned} P_i[k|k-1] &= AP_i[k-1|k-1]A^T + Q[k-1], \\ &\text{node } i \text{ information-update} \end{aligned} \quad (19b)$$

$$I_i[k] = C_i^T R_{(i,i)}^{-1}[k] C_i, \quad i_i[k] = C_i^T R_{(i,i)}^{-1}[k] y_i[k],$$

$$\begin{aligned} &\text{local measurement-update} \\ P_i^{-1}[k|k] &= P_i^{-1}[k|k-1] + \sum_{j \in N_i} I_j[k], \end{aligned} \quad (19c)$$

$$\hat{x}_i^c[k|k] = P_i[k|k] (P_i^{-1}[k|k-1] \hat{x}_i[k|k-1] + \sum_{j \in N_i} i_j[k]),$$

$$\begin{aligned} &\text{local consensus} \\ \hat{x}_i[k|k] &= \hat{x}_i^c[k|k] + \varepsilon \sum_{j \in N_i} (\hat{x}_j^c[k|k] - \hat{x}_i^c[k|k]). \end{aligned} \quad (19d)$$

Due to the ‘‘local-consensus’’-stage this method requires more communication than the DIF, but it does not necessarily

lead to an improved estimation-error. A drawback is that each node estimates the global state-vector, meaning high processing-demand and data transfer per node. A DKF that overcomes this problem is the distributed Kalman filter with bipartite fusion graphs.

## VIII. DISTRIBUTED KALMAN FILTER WITH BIPARTITE FUSION GRAPHS

Originally, the usage of graphs to show how sensors are related to state estimates in DKFs was employed in [17]. More recently, DKFs with bipartite fusion graphs (DKF-BFG) were presented in [18]. The method assumes that each node is connected only to its neighboring nodes collected in  $N_i$ . Furthermore, a node has its own state-estimate which is only a *part* of the global state-vector. This means that the global state-vector at node  $i$ , i.e.  $x_i^{global}$ , is divided into two parts: a part that is estimated, i.e.  $x_i$ , and a part that is not estimated, i.e.  $d_i$ . The vectors  $x_i$  and  $d_i$  are defined using some transformation-matrices  $\Gamma_i$  and  $S_i$  as follows:

$$\begin{pmatrix} x_i[k] \\ d_i[k] \end{pmatrix} = \begin{pmatrix} \Gamma_i \\ S_i \end{pmatrix} x_i^{global}[k]. \quad (20)$$

Preferably, the states of  $x_i$  are determined by taking those states of  $x_i^{global}$  that have a direct relation with the measurement-vector  $y_i$ . Meaning that  $\Gamma_i$  and  $S_i$  are defined by observation-matrix  $C_i$ . Assume  $I$  is the identity matrix with size equal to the number of states in  $x_i^{global}$ . If the  $j^{th}$  column of  $C_i$  contains non-zero elements, the  $j^{th}$  row of  $I$  is put into  $\Gamma_i$ . If not, the  $j^{th}$  row of  $I$  is put into  $S_i$ . An example of  $C_i$  with its corresponding  $\Gamma_i$  and  $S_i$  is:

$$\begin{aligned} C_i &= \begin{pmatrix} c_{11} & c_{12} & 0 & 0 & c_{15} \\ 0 & c_{22} & 0 & 0 & 0 \end{pmatrix} \Rightarrow \\ \Gamma_i &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, S_i = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \end{aligned} \quad (21)$$

Due to the fact that a node  $i$  estimates a part of the global state-vector, the node also has its own process-model derived from the global one. This is done by using  $\Gamma_i$  and  $S_i$  on the global process-model. The following matrices are defined:  $A_i = \Gamma_i A \Gamma_i^T$ ,  $D_i = \Gamma_i A S_i^T$ ,  $H_i = C_i \Gamma_i^T$  and  $w_i[k] = \Gamma_i w[k]$ . With this, the process-model of node  $i$  becomes:

$$\begin{aligned} x_i[k] &= A_i x_i[k-1] + D_i d_i[k-1] + w_i[k-1], \\ y_i[k] &= H_i x_i[k] + v_i[k]. \end{aligned} \quad (22)$$

The method assumes that the state-vector  $x_i$  is estimated by node  $i$  as  $\hat{x}_i$ , state-vector  $d_i$  is sent by other nodes and is represented by node  $i$  as  $\hat{d}_i$ . What remains is the matrix  $Q_i[k] = E(w_i[k] w_i^T[k])$ .

Now that the characteristics of the DKF-BFG are presented, we proceed with the estimation algorithm. Notice that the algorithmic procedure is actually based on the the DIF algorithm in (9). Each node shares its local information-matrix  $I_i$  and information-vector  $i_i$  with its neighbors in  $N_i$ . But because the state-vectors in different nodes are not necessarily equal, in contrast with the DIF, the structure of

$I_i$  and  $i_i$  differs per node. This means that  $I_i$  cannot be added to  $I_j$ , as is the case in (9c). This is solved by using  $\Gamma_i$  and  $S_i$  as shown in the algorithm:

node  $i$  prediction-step

$$\begin{aligned} \hat{x}_i[k|k-1] &= A_i \hat{x}_i[k-1|k-1] + D_i \hat{d}_i[k-1], \\ P_i[k|k-1] &= A_i P_i[k-1|k-1] A_i^T + Q_i[k-1], \end{aligned} \quad (23a)$$

node  $i$  information-update

$$I_i[k] = H_i^T R_{(i,i)}^{-1} H_i, \quad i_i[k] = H_i^T R_{(i,i)}^{-1} y_i[k], \quad (23b)$$

local measurement-update

$$\begin{aligned} P_i^{-1}[k|k] &= P_i^{-1}[k|k-1] + \sum_{j \in N_i} (\Gamma_i \Gamma_j^T) I_j[k] (\Gamma_i \Gamma_j^T)^T, \\ \hat{x}_i[k|k] &= P_i[k|k] P_i^{-1}[k|k-1] \hat{x}_i[k|k-1] \\ &\quad + P_i[k|k] \sum_{j \in N_i} (\Gamma_i \Gamma_j^T) i_j[k] (\Gamma_i \Gamma_j^T)^T. \end{aligned} \quad (23c)$$

An important issue in the performance of this method is whether the global process-model is sparse and localized so that the node's process-model can be derived without loss of generality. If this is indeed the case, its performance should be equal to the DIF. A drawback is that although only local communication is assumed in [18], it is also assumed that the states of  $\hat{d}_i$  are sent by other nodes. This means that extended or even global communication may still be needed. A benefit of this method is that a node only estimates a part of the global state-vector so that its processing-demand per node is low.

## IX. APPLICATION EXAMPLE

This section assess the non-centralized Kalman filters presented in this paper in terms of state-estimation error, communication requirements and robustness against data loss or node break down.

The benchmark process is the heat transfer of a bar. The bar is divided into 100 segments and the temperature  $T_n$  of each segment  $n$  is estimated. The state-vector of the the global process is therefore  $x = (T_1, T_2, \dots, T_{100})^T$ . The bar is heated at the 48<sup>th</sup> segment. The WSN consists of 5 nodes, placed at segment 11, 31, 51, 71 and 91. Each node measures the temperature of its own specific segment. Several of the DKFs are used to estimate the temperature at all 100 segments. A graphical description of this system is shown in Figure 1.

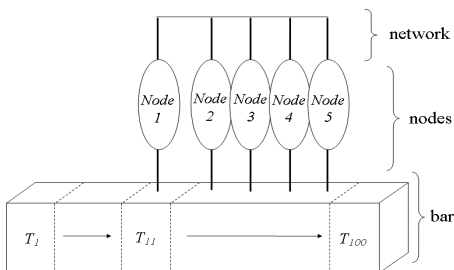


Fig. 1. Bar with Wireless Sensor Network.

The DKFs are first initialized. The sampling time is 10 seconds and the model runs for 10,000 seconds. The initial

state-vector and error-covariance together with  $Q$  and  $R$  are the same for all methods. This concludes the design of the PIF and the DHKF. Communication is only allowed with the neighboring nodes. For example, node 3 receives from and sends data to node 2 and 4. In this way the design of the DIF is also completed. For the DKF-CF the value of  $\epsilon$  is set 0.1, which gave good simulation-results. The design of this parameter is critical, for if too big the estimation algorithm becomes unstable, while if too little the method has no improvements over the DIF algorithm. Matrices  $\Gamma_i$  and  $S_i$  of the DKF-BFG are constructed in such a way that node 1 estimates state 1 to 21, node 2 state 1 to 41, node 3 state 21 to 61, node 4 state 41 to 81 and node 5 state 61 to 100.

Figure 2 and Figure 3 show the real temperature of all the states together with the measurements (with noise) both at 10,000 seconds. Also the estimated states of the different methods are plotted. The estimation of a state's nearest node is plotted, i.e. the plotted states 41 to 60 were estimated by node 3. In case of Figure 2 no data loss was simulated. In Figure 3 however, we simulated a 5% loss of the communicated data-packages.

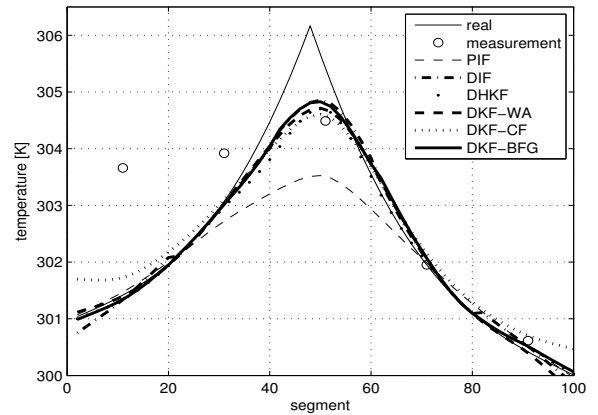


Fig. 2. State-estimation at time 10,000 seconds without data loss.

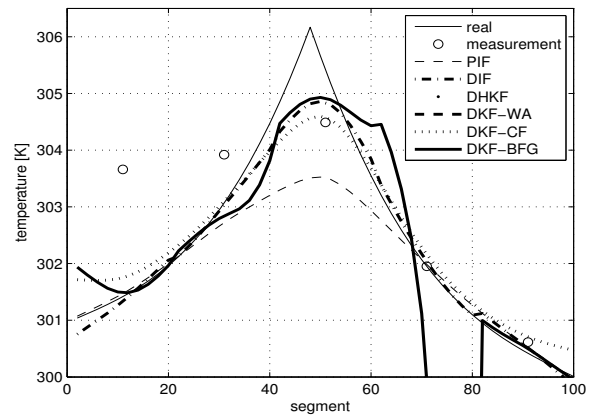


Fig. 3. State-estimation at time 10,000 seconds with 5% data loss.

Beside state-estimation, communication is also an im-

portant aspect. Table I shows which variables need to be transmitted and whether they are transmitted locally (i.e. to node in  $N_i$ ) or globally (i.e. to all nodes in  $N$ ). The total number of sent items is shown in the fourth column. Take for example DIF;  $i_i$  has 100 items and  $I_i$  10,000 items. Nodes 2, 3 and 4 send this data to 2 other nodes which leads to 20,200 items to be sent per node. Nodes 1 and 5 send to 1 other node, resulting in 10,100 sent items per node.

TABLE I  
REQUIRED COMMUNICATION

DKF	variables	nodes	send items per node
PIF	$\hat{x}_i[k k] P_i[k k]$	$N$	40,400
DIF	$i_i[k] I_i[k]$	$N_i$	10,100 or 20,200
DHKF	$\hat{x}_i[k k-1] P_i[k k-1]$ $\hat{x}_i[k k] P_i[k k]$	$N$	3360
DKF-WA	$\hat{x}_i[k]$	$N_i$	200
DKF-CF	$i_i[k] I_i[k] \hat{x}_i^c[k k]$	$N_i$	20,400
DKF-BFG	$i_i[k] I_i[k] / \hat{x}_i[k k]$	$N_i/N$	500 or 1800 or 3440

Figure 2 and Figure 3 together with Table I show the performance, robustness to data loss and the communication requirement, respectively, for each method. Unfortunately, the methods that require the least data transfer, i.e. DKF-WA and DHKF, suffer the most from data loss. Note that the estimated temperature values obtained with these two methods do not even appear in Figure 3 (they are around 100K). Furthermore, also the DKF-BFG estimator, although it needs much less communication than the DIF estimator, in the presence of data loss is not robust, as can be observed in Figure 3. On the overall, the least estimation error was obtained for the DIF estimator, which is also the most robust against data loss. Another aspect that can be observed is that the process-model is almost localized and sparse, as the results of the DKF-BFG closely resemble the ones obtained with the DIF, when no data loss occurs.

## X. CONCLUSIONS

In this paper we presented an overview of different methodologies for designing non-centralized Kalman filters that can be used in WSNs. Each method was described and analyzed in terms of communication requirements, robustness and estimation-error. It turned out that the DKF-WA requires the least communication and provides a low state-estimation error. However, it lacks robustness for its estimation error increases significantly when data is lost or nodes break down, which is usually the case in WSNs. For this reason it is not suitable for most WSNs. A method that can deal with unreliable data transfer and node loss, but still has a low state-estimation error is the DIF. It also has average requirements regarding the amount of data transfer needed compared to other methods. The amount of computations and communication per node can be decreased when the DKF-BFG is used. However, this approach is valid only for processes that have a localized and sparse structure, and assuming that there is no data-loss. Hence, the DKF-BFG is not suitable for usage in WSNs.

An extension on this survey paper is to take mathematical models for communication into account. Meaning that both

communication topology as well as the introduced errors and noises due to wireless communication links are used in the noise- and stability analysis, as described in [19].

Based on the above conclusions, future work on non-centralized estimators, suitable for WSNs, needs to find new methods for reducing the communication and computation requirements, without loosing robustness to data loss. Improving the robustness of the DKF-BFG seems to be a possible solution.

## REFERENCES

- [1] R. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME Journal of Basic Engineering*, vol. 82, no. D, pp. 35–42, 1960.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a survey," *Elsevier, Computer Networks*, vol. 38, pp. 393–422, 2002.
- [3] J. Speyer, "Computation and transmission requirements for a decentralized Linear-Quadratic-Gaussian control problem," *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, 1979.
- [4] S. Felter, "An overview of decentralized Kalman filters," in *IEEE 1990 Southern Tier Technical Conference*, Birmingham, NY, USA, 1990, pp. 79–87.
- [5] S. Roy, R. Hashemi, and A. Laub, "Square root parallel Kalman filtering using reduced order local filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 2, pp. 276–289, 1991.
- [6] S. Roy and R. Iltis, "Decentralized linear estimation in correlated measurement noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 6, pp. 939–941, 1991.
- [7] S. Shu, "Multi-sensor optimal information fusion Kalman filters with applications," *Elsevier, Aerospace Science & Technology*, vol. 8, no. 1, pp. 57–62, 2004.
- [8] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *4th Int. Symp. on Information processing in sensor networks*, Los Angeles, California, USA, 2005.
- [9] H. Durant-Whyte, B. Rao, and H. Hu, "Towards a fully decentralized architecture for multi-sensor data fusion," in *1990 IEEE Int. Conf. on Robotics and Automation*, Cincinnati, Ohio, USA, 1990, pp. 1331–1336.
- [10] H. Hashmipour, S. Roy, and A. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–93, 1988.
- [11] M. Hassan, G. Salut, M. Sigh, and A. Titli, "A decentralized algorithm for the global Kalman filter," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 262–267, 1978.
- [12] R. Quirino and C. Bottura, "An approach for distributed Kalman filtering," *Revista Controle & Automatica da Sociedade Brasileira de Automática*, vol. 21, pp. 19–28, 2001.
- [13] P. Alriksson and A. Rantzer, "Distributed Kalman filter using weighted averaging," in *Proc. of the 17th Int. Symp. on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.
- [14] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *46th IEEE Conf. on Decision and Control*, New Orleans, LA, USA, 2007.
- [15] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *American Control Conference*, Albuquerque, New Mexico, 1997.
- [16] R. Olfati-Saber, "Distributed Kalman filter using embedded consensus filters," in *44th IEEE Conf. on Decision and Control 2005 and 2005 European Control Conference (CDC-ECC'05)*, Seville, Spain, 2005, pp. 8179–8184.
- [17] A. Mutambara and D.-W. H.F., "Fully decentralized estimation and control for a modular wheeled mobile robot," *International Journal of Robotic Research*, vol. 19, no. 6, pp. 582–596, 2000.
- [18] U. Khan and J. Moura, "Distributed Kalman filters in sensor networks: Bipartite Fusion Graphs," in *IEEE 14th Workshop on Statistical Signal Processing*, Madison, Wisconsin, USA, 2007, pp. 700–704.
- [19] R. Smith and F. Hadaegh, "Closed-Loop Dynamics of Cooperative Vehicle Formations With Parallel Estimators and Communication," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1404–1414, 2007.