

Optimal decentralized Kalman filter

Citation for published version (APA):

Oruç, S., Sijs, J., & Bosch, van den, P. P. J. (2009). Optimal decentralized Kalman filter. In *Proceedings of the 17th Mediterranean Conference on Control and Automation, MED '09, 24-26 June 2009, Thessaloniki, Greece*, (pp. 803-808). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/MED.2009.5164642>

DOI:

[10.1109/MED.2009.5164642](https://doi.org/10.1109/MED.2009.5164642)

Document status and date:

Published: 01/01/2009

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

Take down policy

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

openaccess@tue.nl

providing details and we will investigate your claim.

Optimal Decentralized Kalman Filter

S. Oruç, J. Sijs, P.P.J. van den Bosch

Abstract—The Kalman filter is a powerful state estimation algorithm which incorporates noise models, process model and measurements to obtain an accurate estimate of the states of a process. Implementation of conventional Kalman filter algorithm requires a central processor that harvests measurements from all the sensors in the field. Central algorithms have some drawbacks such as reliability, robustness and high computation which result in a need for non-central algorithms. This study takes optimality in Decentralized Kalman Filter (DKF) as its focus and derives the Optimal Decentralized Kalman Filter (ODKF) algorithm, in case the network topology is provided to every node in the network, by introducing *Global Kalman Equations*. ODKF sets a lower bound of estimation error in least squares sense for DKF.

Index Terms—State estimation, sensor network, decentralized Kalman filter

I. INTRODUCTION

The Kalman filter [1] combines information from the model and the real time measurements of a process to estimate the states of the process. In its original form, the Kalman filter requires all the measurements to be sent to a central processor for estimation. Later on research interest in this field has been directed to decentralize the Kalman filter to apply it in distributed systems like Distributed Sensor Networks (DSN) consisting of large amount of nodes.

The early papers suggested multi sensor Kalman filtering schemes with hierarchical but decentralized structures. [2]–[4]. Later on the *Information filter*, which is a non-hierarchical decentralized Kalman filter, is introduced where all sensors work in parallel to obtain local estimation based on their own and neighbors' information [5]–[9]. These algorithms are optimal given the constraint of sharing only *measurements*. We refer to such optimal algorithms, where only measurements are shared but not state estimations, as Local Kalman Filter (LKF) in this paper. Contrary to LKF, Decentralized Kalman Filter (DKF) does share *state estimations* as well as *measurements* which create more complex problems. does not let the nodes share *state estimations* which creates more complex problems. By transmitting *state estimation*, the past measurements of each node can diffuse throughout the network. The importance of transmitting state estimation is illustrated in Temperature Distribution example.

S. Oruç is with the Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, E-mail: sertaco@gmail.com.

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

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

This paper aims to introduce an optimal formulation and infer an optimal algorithm for DKF which we call ODKF.

The more recent papers in this field mention the agreement protocols between the nodes based on estimated states [10]–[12]. These algorithms provide non-optimal but practically acceptable solutions.

II. KEY CONCEPTS AND NOTATION

Node

Unit which has a sensor to make measurements, a processor to run an algorithm and a network connection to neighboring nodes.

Neighbor

The node that is communication-wise adjacent to the node under consideration.

Neighborhood

The set of nodes that is adjacent to the node under consideration and the node itself.

Optimality

Minimization of the sum of squared errors between state estimate, \hat{x}_i and the real state x according to the model.

Data Incest

The situation in which the same data is used more than once in the estimation.

$x[k]$	The real state at the k^{th} iteration.*
$y_i[k]$	The measurement of node i at the k^{th} iteration.
\hat{x}_i	The estimate of state x of node i .
\hat{x}_i^-	The predicted estimate of state x of node i .
$P_{ij}[k]$	$E[(x[k] - \hat{x}_i[k])(x[k] - \hat{x}_j[k])^T]$.
M_{ij}	Kalman gain for state from j^{th} node on i^{th} node.
K_{ij}	Kalman gain for measurement from j^{th} node on i^{th} node.
N	The set of all nodes in the network.
N_i	Neighborhood of i^{th} node.
N_i^-	Neighborhood of i^{th} node w/o itself i.e. $N_i - \{i\}$.
v_i	Measurement-noise of the i^{th} sensor.
w	Process-noise.
R_i	$E[v_i v_i^T]$.
Q	$E[ww^T]$.

*For clarity time indices are dropped when there is no risk of confusion.

III. PROBLEM DESCRIPTION

The system we consider in this paper is a network of *nodes* with their own sensor, hence own measurements. The network topology is arbitrary but known by each node. An example of a certain network topology and process dynamics is shown in figure 1. There is only one process and each

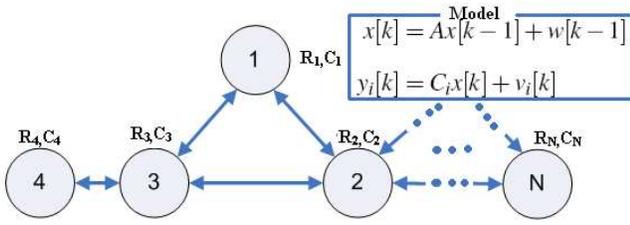


Fig. 1. System

node estimates the entire state-vector of the process based on its own and neighboring nodes' measurement and previous state estimation. Therefore the discrete state-space model of a local node i becomes:

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

$$y_i[k] = C_i x[k] + v_i[k]. \quad (1b)$$

The probability density functions (PDFs) of w and v are modeled as zero mean Gaussian noises with an error-covariance matrix of Q and R_i respectively. The sensor measurements are assumed to be uncorrelated.

With this network of sensors we want to estimate the state-vector x of the process. The optimal estimator for the described process of (1) is the Kalman filter [1] for which all measurements are gathered by a central algorithm. Therefore, if we define $y := [y_1^T, y_2^T, \dots, y_n^T]^T$ and $v := [v_1^T, v_2^T, \dots, v_n^T]^T$, then the process model of the Kalman filter becomes:

$$x[k] = Ax[k-1] + w[k-1] \quad (2)$$

$$y[k] = Cx[k] + v[k] \quad (3)$$

The Kalman filter's update equations, with $R := E[vv^T] = \text{diag}(R_1, R_2, \dots, R_n)$, yields:

$$\hat{x}^-[k] = A\hat{x}[k-1] \quad (4)$$

$$P^-[k] = AP[k-1]A^T + Q \quad (5)$$

$$K[k] = P^-[k]C^T(CP^-[k]C^T + R)^{-1} \quad (6)$$

$$\hat{x}[k] = \hat{x}^-[k] + K[k](y[k] - C\hat{x}^-[k]) \quad (7)$$

$$P[k] = (I - K[k]C)P^-[k] \quad (8)$$

Although this estimator is optimal it requires that every measurement is sent to one central processor, implying that it cannot be used in the system of Figure 1. As a result of that, it does not make sense to compare an estimator that is designed for a sensor network with the central Kalman filter since the latter one does not take the communication topology of the network into account. Nevertheless, a lot of DKFs make this comparison. Therefore, the goal of this paper is to present an optimal state-estimator, i.e. ODKF, which takes explicitly the network topology into account. Then future DKFs can compare their results with the ODKF to assess the performance of their estimator.

For a DKF the most important parameter is the estimation-error i.e. $x[k] - \hat{x}_i[k]$. In case of the Kalman filter, its PDF is modeled as a Gaussian function. It can be proven that with the equations of the Kalman filter this PDF has a zero mean, i.e. $E[x[k] - \hat{x}_i[k]] = 0$, and a covariance modeled as

$P_{ii}[k]$. The aim of any DKF is to minimize the total value of all $P_{ii}[k]$ given the communication requirements. Therefore, the ODKF minimizes trace of $\sum_{i=1}^n P_{ii}[k]$ under the standing assumption that the network topology is known and that any parameter of the node's local state-estimator can be sent to its neighboring nodes.

The designed ODKF is compared with the local Kalman filter (LKF) in an application example on the temperature distribution of a bar. Therefore, before going into the details of the ODKF, let us first present the LKF.

IV. LOCAL KALMAN FILTER

The local Kalman filter of [5], [8] is based on the Kalman filter in information form. In this form, each measurement of node i , i.e. y_i , and its covariance R_i are transformed into an information-vector i_i and information-matrix I_i :

$$i_i[k] := C_i^T R_i^{-1} y_i[k], \quad (9)$$

$$I_i := C_i^T R_i^{-1} C_i. \quad (10)$$

The LKF does not take the communication topology into account, it just assumes that there are neighboring nodes. Each node shares its local information-vector i_i and corresponding information-matrix I_i with its neighbors. Therefore node i uses the received information of all the nodes in the set N_i to update its local state-vector \hat{x}_i and error-covariance matrix P_{ii} . For that node i applies the following equations:

$$P_{ii}[k]^- = AP_{ii}[k-1]A^T + Q, \quad (11)$$

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

$$P_{ii}[k] = \left((P_{ii}[k]^-)^{-1} + \sum_{j \in N_i} I_j \right)^{-1}, \quad (13)$$

$$\hat{x}_i[k] = P_{ii}[k] \left((P_{ii}[k]^-)^{-1} \hat{x}_i[k]^- + \sum_{j \in N_i} i_j[k] \right). \quad (14)$$

This algorithm is proven to give optimal estimate for i^{th} node given the constraint that only the measurements are communicated to neighboring nodes and are not forwarded to other nodes.

V. OPTIMAL DECENTRALIZED STATE-ESTIMATOR

Contrary to the LKF, in the ODKF each node transmits both its local state estimate \hat{x}_i and measurement y_i to its neighbors. Note that the measurements are fresh data which are not correlated with the current estimates or other measurements made in the neighborhood. The estimates however are found by incorporation of historical information, therefore they are correlated with the estimates from the other nodes. Furthermore, for the LKF holds that measurements made at a second order neighbor of node i can never have an effect on the estimation of x_i . However, if the nodes communicate the state estimates as well as the measurements, as in the ODKF. Then inherently the information of measurements made at a second order neighbor of node i does have effect on the estimation of x_i through the estimated state of a first order neighbor.

The transmission of estimated states to neighbors brings in the risk of data incest since the estimated states of different nodes can be correlated. To prevent data incest, we assume that the network topology is known by each node and taken into account in the derivation of ODKF.

As stated in Section III, the aim of the ODKF is to minimize the trace of $\sum_{i=1}^n P_{ii}[k]$. However, calculation of $P_{ii}[k]$ is not straight forward since $\hat{x}_i[k]$ involves the incorporation of $\hat{x}_j[k-1]$, for all $j \in N_i$. Therefore to calculate $P_{ii}[k]$ we need $P_{ij}[k-1]$, since $P_{ij}[k-1]$ is defined as $P_{ij}[k-1] = E[(x[k-1] - \hat{x}_i[k-1])(x[k-1] - \hat{x}_j[k-1])^T]$. Calculation of these P_{ij} terms is the key challenge in finding the ODKF algorithm.

Our approach in the design of the ODKF is a simple yet effective one. The cross-covariances are specifically taken into account by considering the network as a whole. For this purpose we introduce a new representation, called *Global System Representation*, as presented in Section V-A. This representation enables us to calculate all cross covariances $P_{ij}[k]$ in the network, from which a cost-function can be derived to minimize the trace of $\sum_{i=1}^n P_{ii}[k]$ as presented in Section V-B.

A. Global Kalman Equations

In this paper we aim to take the cross correlations, i.e. P_{ij} , between the estimates of each node explicitly into account as well as the auto correlations, i.e. P_{ii} . For clarity let us start with the state space model of the process at node i :

$$x[k] = Ax[k-1] + w[k-1], \quad (15)$$

$$y_i[k] = C_i x[k] + v_i[k]. \quad (16)$$

Then the process for the entire network can be expressed as;

$$x[k] = Ax[k-1] + w[k-1], \quad (17)$$

$$Y[k] = \Gamma x[k] + V[k]. \quad (18)$$

with the following definitions:

$$Y := \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \Gamma := \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix}, V := \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

In the derivation of the Kalman filter [1] it is defined that the update of the state-vector is $x[k] = x^-[k] + K[k](y[k] - Cx^-[k])$. Then the optimal solution of the estimation was derived by substitution of that $K[k]$ which solves the equation $\frac{\delta Tr(P[k])}{\delta K[k]} = 0$. Applying the same derivation here, gives:

$$\begin{aligned} \hat{x}_i[k] &= \hat{x}_i^-[k] + \sum_{j \in N_i} K_{ij}[k](y_j[k] - C_j \hat{x}_i^-[k]) \\ &+ \sum_{j \in N_i^-} M_{ij}[k](\hat{x}_j^-[k] - \hat{x}_i^-[k]), \end{aligned} \quad (19)$$

$$= \sum_{j \in N_i} M_{ij}[k] \hat{x}_j^-[k] + \sum_{j \in N_i} K_{ij}[k] y_j[k], \quad (20)$$

$$\text{where } M_{ii} := I - \sum_{j \in N_i^-} M_{ij} - \sum_{j \in N_i} K_{ij} C_j. \quad (21)$$

Here the predicted state \hat{x}_i^- is corrected with the measurement and state residue terms $y_j - C_j \hat{x}_i^-$ and $\hat{x}_j^-[k] - \hat{x}_i^-[k]$ respectively. These residue terms are weighed with their corresponding Kalman gains K_{ij} and M_{ij} . Then;

$$\begin{aligned} y_i &= C_i x + v_i, \\ e_i &= x - \hat{x}_i = \sum_{j \in N_i} M_{ij} e_j^- - \sum_{j \in N_i} K_{ij} v_j, \end{aligned} \quad (22)$$

$$e_i = \mu_i \begin{bmatrix} e_1^- \\ e_2^- \\ e_3^- \\ \vdots \end{bmatrix} - \kappa_i \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \end{bmatrix}, \quad (23)$$

$$P_{ii} = \mu_i \begin{bmatrix} P_{11}^- & P_{12}^- & \cdots \\ P_{21}^- & P_{22}^- & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \mu_i^T + \kappa_i \begin{bmatrix} R_1 & 0 & \cdots \\ 0 & R_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \kappa_i^T. \quad (24)$$

Making the following definitions;

$$\begin{aligned} \hat{X} &:= \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix}, \Delta := \begin{bmatrix} A & 0 & 0 & \cdots \\ 0 & A & 0 & \cdots \\ 0 & 0 & A & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \Phi := \begin{bmatrix} Q & Q & Q & \cdots \\ Q & Q & Q & \cdots \\ Q & Q & Q & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\ \Pi &:= \begin{bmatrix} P_{11} & P_{12} & P_{13} & \cdots \\ P_{21} & P_{22} & P_{23} & \cdots \\ P_{31} & P_{32} & P_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \kappa := \begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots \\ K_{21} & K_{22} & K_{23} & \cdots \\ K_{31} & K_{32} & K_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \\ \mu &:= \begin{bmatrix} M_{11} & M_{12} & M_{13} & \cdots \\ M_{21} & M_{22} & M_{23} & \cdots \\ M_{31} & M_{32} & M_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \Omega := \begin{bmatrix} R_1 & 0 & 0 & \cdots \\ 0 & R_2 & 0 & \cdots \\ 0 & 0 & R_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \end{aligned}$$

The *Global Kalman Equations* without optimal Kalman Gains κ, μ is found as:

$$\text{predict: } \hat{X}^-[k] = \Delta \hat{X}[k-1], \quad (25)$$

$$\Pi^-[k] = \Delta \Pi[k-1] \Delta^T + \Phi, \quad (26)$$

$$\text{update: } \hat{X}[k] = \mu[k] \hat{X}^-[k] + \kappa[k] Y[k], \quad (27)$$

$$\Pi[k] = \mu[k] \Pi[k]^- \mu[k]^T + \kappa[k] \Omega \kappa[k]^T. \quad (28)$$

The trace of $\Pi[k]$ equals the cost-function in our optimization problem. Meaning that our aim is to find those κ and μ which minimize the trace of equation (28).

The network connections are explicitly taken into account in deriving equation (28) such that K_{ij} and M_{ij} are defined as the zero-matrix in case there is no communication between i^{th} and j^{th} node. Note that M_{ij} , for all $i \neq j$, and K_{ij} are independent variables whereas M_{ii} depends on these variables as in equation (21).

B. Optimal Decentralized Kalman filter

In order to find the ODKF equations we need to minimize $Tr(\Pi)$ for the Kalman Gains K_{ij} and M_{il} with $i \in N, j \in N_i$ and $l \in N_i^-$. If for clarity P_{ii} is replaced with P_{mm} , then with

the equality $Tr(\Pi[k]) = \sum_{m=1}^N Tr(P_{mm})$, this cost-function becomes:

$$\min_{K_{ij}, M_{il}} (Tr(\Pi)) = \min_{K_{ij}, M_{il}} \sum_{m=1}^N Tr(P_{mm}). \quad (29)$$

On the other hand we have for P_{mm} , after substitution of the definitions of κ_i, μ_i into (24),

$$P_{mm} = \sum_{j \in N_m} \sum_{l \in N_m} M_{ml} P_{lj}^- M_{mj}^T + \sum_{j \in N_m} K_{mj} R_j K_{mj}^T. \quad (30)$$

Due to the fact that all P_{ij}^- 's are constants coming from previous iteration, it follows from (30) that only the P_{mm} for which $i = m$ depends **only** on K_{ij} and M_{il} . All the other ones, i.e. P_{mm} for all $m \in N \setminus i$ do not depend on K_{ij} and M_{il} . As a result minimization of the cost-function as shown in (29) becomes:

$$\min_{K_{ij}, M_{il}} (Tr(\Pi)) = \min_{K_{ij}, M_{il}} Tr(P_{ii}). \quad (31)$$

So to minimize the ‘‘global’’ cost-function it is enough to minimize individual P_{ii} 's with respect to their corresponding K_{ij}, M_{il} . The values for K_{ij} and M_{il} , which minimize the cost-function, are those values at which the derivative of P_{ii} equals 0, i.e.:

$$\frac{\delta}{\delta K_{ij}} Tr(P_{ii}) = -2 \sum_{l \in N_i} M_{il} P_{li}^- C_j^T + 2 K_{ij} R_j = 0, \quad j \in N_i, \quad (32)$$

$$\frac{\delta}{\delta M_{ij}} Tr(P_{ii}) = -2 \sum_{l \in N_i} M_{il} (P_{li}^- - P_{lj}^-) = 0, \quad j \in N_i^-. \quad (33)$$

Rewriting equations (32) and (33), by replacing

$$M_{ii} = I - \sum_{l \in N_i^-} M_{il} - \sum_{l \in N_i} K_{il} C_l$$

as in equation (21), gives:

$$\begin{aligned} P_{ii}^- C_j^T &= \sum_{l \in N_i^-} M_{il} (P_{li}^- - P_{lj}^-) C_j^T \\ &+ \sum_{l \in N_i} K_{il} C_l P_{li}^- C_j^T + K_{ij} R_j, \quad \text{where } j \in N_i, \end{aligned} \quad (34)$$

$$\begin{aligned} (P_{ii}^- - P_{ij}^-) &= \sum_{l \in N_i^-} M_{il} (P_{li}^- - P_{lj}^-) + \sum_{l \in N_i} K_{il} C_l (P_{li}^- - P_{lj}^-) \\ &- \sum_{l \in N_i^-} M_{il} (P_{li}^- - P_{lj}^-), \quad \text{where } j \in N_i^-. \end{aligned} \quad (35)$$

This set of equations gives $2n - 1$ matrix equations from which the Kalman gains are calculated, where n is the number of nodes in N_i . Suppose we define:

$$\kappa_i := [K_{i1} \ K_{i2} \ K_{i3} \ \cdots \ K_{ij} \ \cdots], \quad \text{where } j \in N_i,$$

$$\mu_i := [M_{i1} \ M_{i2} \ \cdots \ M_{ij} \ \cdots], \quad \text{where } j \in N_i,$$

$$\Gamma_i := [C_1^T, C_2^T, \dots, C_j^T, \dots]^T, \quad \text{where } j \in N_i.$$

Then (34) and (35), for all nodes i , can be written in matrix form as;

$$[\kappa_i \ \mu_i] \begin{bmatrix} \mathfrak{A}_i & \mathfrak{B}_i \\ \mathfrak{B}_i^T & \mathfrak{C}_i \end{bmatrix} = [P_{ii}^- \Gamma_i^T \ \mathbb{P}_{ii}]. \quad (36)$$

where

$$\mathfrak{A}_i = \Gamma_i P_{ii}^- \Gamma_i^T + \Omega_i, \quad (37)$$

$$\mathfrak{B}_i = \Gamma_i \mathbb{P}_{ii}, \quad (38)$$

$$\mathfrak{C}_i = \begin{bmatrix} \mathbb{P}_{ii} \\ \mathbb{P}_{ii} \\ \vdots \\ \mathbb{P}_{ii} \\ \vdots \end{bmatrix} - \begin{bmatrix} \mathbb{P}_{1i} \\ \mathbb{P}_{2i} \\ \vdots \\ \mathbb{P}_{ki} \\ \vdots \end{bmatrix}, \quad (39)$$

$$\mathbb{P}_{ji} = [P_{ji}^- \ P_{ji}^- \cdots P_{ji}^- \cdots] - [P_{j1}^- \ P_{j2}^- \cdots P_{jk}^- \cdots], \quad (40)$$

where $j \in N_i$ and $k \in N_i^-$.

At this point it could be helpful for the reader to note that to write equations (39) and (40), we assume node 1 and 2 are neighbors of i^{th} node for the sake of notation.

Hence the optimal DKF at i^{th} node yields:

predict:

$$\hat{x}_i^- [k] = A \hat{x}_i [k-1] \quad (41)$$

$$P_{ij}^- [k] = A P_{ij} [k-1] A^T + Q \quad (42)$$

Determination of Kalman gains κ_i, μ_i

$$[\kappa_i \ \mu_i] = [P_{ii}^- \Gamma_i^T \ \mathfrak{B}_i] \begin{bmatrix} \mathfrak{A}_i & \mathfrak{B}_i \\ \mathfrak{B}_i^T & \mathfrak{C}_i \end{bmatrix}^{-1} \quad (43)$$

update:

$$\begin{aligned} \hat{x}_i [k] &= \hat{x}_i^- [k] + \sum_{j \in N_i} K_{ij} [k] (y_j [k] - C_j \hat{x}_i^- [k]) \\ &+ \sum_{j \in N_i} M_{ij} [k] (\hat{x}_j^- [k] - \hat{x}_i^- [k]) \end{aligned} \quad (44)$$

$$\Pi [k] = \mu [k] \Pi [k]^- \mu [k]^T + \kappa [k] \Omega \kappa [k]^T \quad (45)$$

Notice that the covariance matrices, which are required for $\mathfrak{A}_i, \mathfrak{B}_i$ and \mathfrak{C}_i , are calculated with their governing equation (45). It involves global variables which are all the predicted auto and cross covariance matrices Π^- as well as all the Kalman gains κ, μ in the network. The covariance matrices and the Kalman gains depend on the measurement's covariance R_i and not on the actual measurements y_i since they are calculated with (42), (43) and (45). Therefore Π, κ and μ can be calculated in each local node if the nodes know the network topology. To build κ and μ each node first calculates κ_i and μ_i for all $i \in N$ using (43) which gives non zero entries for κ and μ matrices. And then the rest of the elements in κ and μ are assigned to zero, which corresponds to the unconnected nodes in the network.

Remark: Depending on the network topology, some or all states in the different nodes converge to the same value. As a result all cross covariance matrices, i.e. P_{ij} , will also converge to the same value making \mathfrak{B}_i and \mathfrak{C}_i singular and thus making the inversion of (43) impossible. This is an inherent point that stems from the formulation of Π and \hat{X} when the augmented state was defined. As the states of different nodes converge to each other, $\hat{x}_j - \hat{x}_i$ goes to zero and M_{ij} of (45) might have any value. Occurrence of singularity depends on the precision of the processor.

Next, an example of the temperature distribution of a bar is used to compare the ODKF, the LKF and the central Kalman filter (CKF) as well as revealing the singularity issue.

VI. EXPERIMENT: TEMPERATURE DISTRIBUTION

To illustrate the performance of the ODKF algorithm, as proposed in Section V-B, we simulate an experiment in which a bar is connected to two temperature reservoirs on both ends and heated from the middle. Our purpose is comparing the ODKF with both the CKF and LKF. The CKF is done by a central processor which accesses all four measurements directly. Both the LKF and DKF are run in the local processors at the nodes.

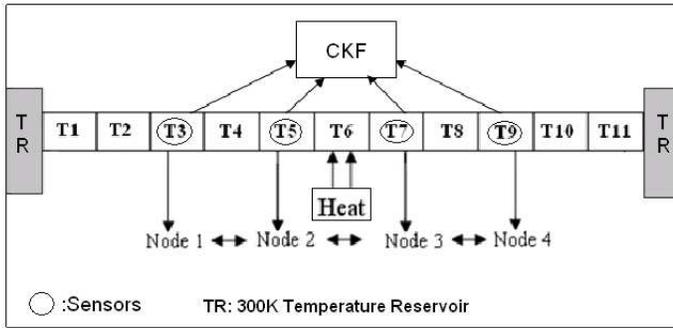


Fig. 2. Setup of the measurement system.

The bar is divided into 11 segments and therefore modeled with 11 states each representing the temperature of one segment of the bar, as shown in figure 2. The reservoirs provide boundary conditions for the experiment and keep the temperature of the end points at 300 K. The bar is heated from the 6th segment and the temperatures of only the 3rd, 5th, 7th and 9th segments are measured. Each measurement is done by a node and the nodes can communicate with their neighbors to collaborate data. Initially the bar is kept at 300K which changes in time due to the heat at the 6th segment. The temperature profile of the bar over time is shown in figure 3 with solid line.

All three state-estimators use the same state-space model of the system as in figure 1, i.e. $x[k] = Ax[k-1] + w[k-1]$ and $y_i[k] = C_i x[k] + v_i[k]$. The state-vector x contains 11 elements, each representing the temperature of one segment and w also contains 11 elements, each representing the unknown heating/cooling at one of the segments. Also C_i denotes measurement matrix and v_i measurement noise. Notice that in this example the first node has a measurement matrix $C_1 = [00100000000]$. The nodes do not know about the heating/cooling given to the bar, nor about the heat-reservoirs at the bar's ends and models those as Gaussian noise. Therefore the covariance of the process-noise, i.e. Q , is quite high with respect to R_i , meaning that the measurements are more "reliable" than the process-model.

The reason for using this setup is that since the uncertainty in the process model is very high, the measurements are most valuable for state-estimation. In our system, although we have 11 segments we only measure 4 of them while the

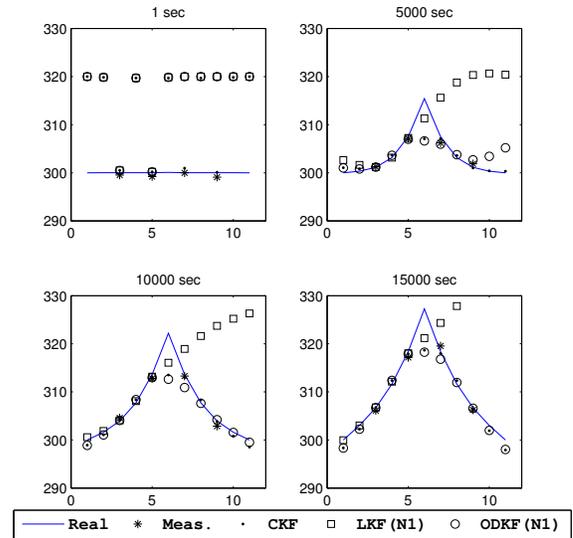


Fig. 3. Temperature profile of the bar over time.

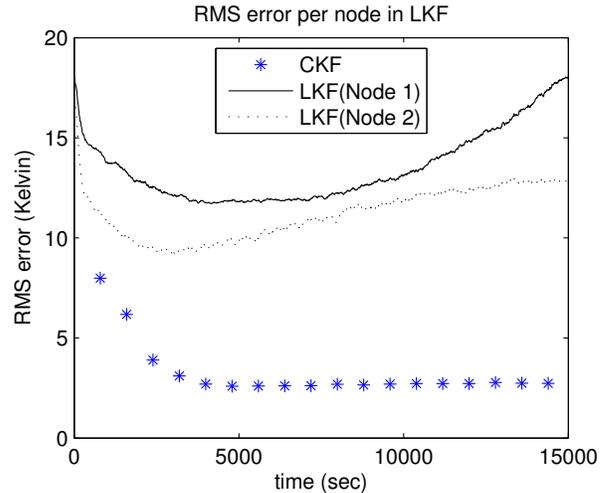


Fig. 4. Local Kalman Filter

rest is to be estimated from the model. That is to say, this setup contains a lot of states that are not measured at all but are yet to be estimated. The comparison between the LKF and the DKF illustrates the importance of sharing states. For example in the LKF the 1st node can only use measurements from the 1st and the 2nd node. Therefore its estimation-error of states on the right side of the bar is high. However in the ODKF sharing states results in the diffusion of measurements from the 3rd and 4th node inherently, which results in a much better estimation as illustrated in figure 3.

The performance of the LKF and the ODKF are compared with the CKF, as the CKF is assumed to be state-estimator with the best overall performance, i.e. the least estimation-error. The result of the LKF are presented in Figure 4 and of the ODKF in Figure 5. This figure suggest that the ODKF may converge to the CKF, which is left as a future work.

Figure 4 shows the root-mean-squared (RMS) error in time

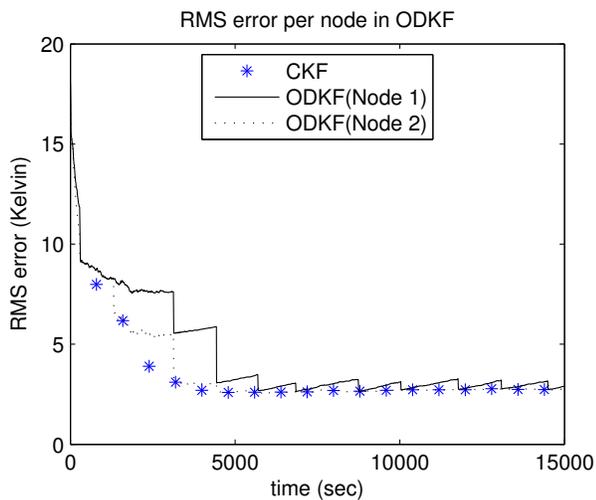


Fig. 5. Optimal Decentralized Kalman Filter

of the LKF between the real states and estimated ones at 1st and 2nd node. Since the process and the node's position are symmetric from the center of the bar, the error of the 3rd and 4th nodes follow more or less the same trajectories as the ones of the 2nd and 1st node respectively. For this reason these two nodes are not drawn for clarity. The LKF of the 1st node has access to the measurements from the 2nd node and itself. Because of this reason its estimation is worse than 2nd node, for that node has access to the measurements from the 1st and 3rd node.

In our experiment for the estimation of the temperature distribution by the ODKF, shown in Figure 5, a hybrid algorithm is used. Here both states and measurements are sent and the estimator switches between the ODKF and LKF. When the matrix Π , as denoted in Section V-A, is full rank ODKF algorithm is used. In case Π is singular the LKF is applied. When the LKF algorithm runs, each node makes their own estimation resulting in different error-covariance matrices, P_{ij} . Therefore after a while Π matrix again becomes full rank which triggers the use of ODKF algorithm. This dynamics, which depends on the singularity of Π matrix causes the "saw tooth" graph of Figure 5. This behavior also proves that the ODKF decreases the estimation-error significantly and gives a clue that ODKF may indeed converge to CKF.

In conclusion this experiment is a clear example to show the importance of sending states rather than sending only measurements (as is done in most other state-estimators dedicated to sensor networks). Many network-topology specific solutions [13] can be suggested for the problem. However the aim of this experiment is to compare the ODKF with both the LKF and CKF with respect to already mentioned performance measure. A simple structure is chosen for illustration purposes although it is possible to use an arbitrary and connected graph with ODKF.

VII. CONCLUSION

In this study we addressed some optimality problems arising from the decentralization of the Kalman filter. Considering the network-topology, we stated a problem formulation for optimality of any state-estimator. Solving this problem formulation, we obtained the ODKF in which the network topology is known. This result proposes an algorithm which makes the best estimate that can be obtained, taking the network topology into account. So far in the literature the proposed algorithms for decentralized Kalman filters have been compared with the central KF. However the central KF does not take the network topology into account whereas our ODKF does. For fully connected network topologies it could turn out that the ODKF converges to the central KF, as it is the case for the experiment, which still needs to be proven.

In this paper we examined the optimality condition constrained by the network topology. We did not take its processing demand or data transfer into account. Although this solution of the ODKF is proven to be optimal, since it depends on the network topology, scalability is still an issue to be solved. Meaning that adding or removing a node in the network will affect the algorithms on all nodes since this affects network topology. Therefore future work is on incorporating scalability condition into the problem formulation. The scalability is a key condition to make the algorithm feasible for use in Distributed Sensor Networks.

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] D. Willner, C. Chang, and K. Dunn, "Kalman filter algorithms for a multi-sensor system," *IEEE Conference on Decision and Control Vol.15, Part 1*, vol. 15, no. 1, pp. 570 – 574, 1976.
- [3] M. Hassan, G. Salut, M. Singh, and A. Titli, "A decentralized algorithm for the global Kalman filter," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 262–267, 1978.
- [4] 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.
- [5] H. Durrant-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.
- [6] B. Rao and H. Durrant-Whyte, "Fully decentralized algorithm for multisensor Kalman filtering," *IEEE Proceedings*, vol. 138, no. 5, pp. 413 – 420, 1991.
- [7] S. Roy and R. Ittis, "Decentralized linear estimation in correlated measurement noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 6, pp. 939–941, 1991.
- [8] A. Mutambara and H. Durrant-Whyte, "A formally verified modular decentralized robot control system," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993.
- [9] S. Shu, "Multi-sensor optimal information fusion Kalman filters with applications," *Elsevier, Aerospace Science & Technology*, vol. 8, no. 1, pp. 57–62, 2004.
- [10] R. Saber, "Distributed Kalman filters for sensor networks," in *46th IEEE Conf. on Decision and Control*, New Orleans, LA, USA, 2007.
- [11] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.
- [12] S. Roy, A. Saberi, and K. Herlugson, "A control-theoretic perspective on the design of distributed agreement protocols," 2005.
- [13] S. Stankovic, M. Stanojevic, and D. Siljak, "Decentralized overlapping control of a platoon of vehicles," *IEEE transactions on Control Systems Technology*, 2000.