Technical report 11-028

Decentralized Kalman filter comparison for distributed-parameter systems: A case study for a 1D heat conduction process*

Z. Hidayat, R. Babuška, B. De Schutter, and A. Núñez

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/11_028.html

Decentralized Kalman Filter Comparison for Distributed-Parameter Systems: A Case Study for a 1D Heat Conduction Process

Z. Hidayat, R. Babuška, B. De Schutter, A. Núñez

Delft Center for Systems and Control, Delft University of Technology

Delft, The Netherlands

{z.hidayat, r.babuska, b.deschutter, a.a.nunezvicencio}@tudelft.nl

Abstract

In this paper we compare four methods for decentralized Kalman filtering for distributed-parameter systems, which after spatial and temporal discretization, result in large-scale linear discrete-time systems. These methods are: parallel information filter, distributed information filter, distributed Kalman filter with consensus filter, and distributed Kalman filter with weighted averaging.

These filters are suitable for sensor networks, where the sensor nodes perform not only sensing and computations, but also communicate estimates among each other. We consider an application of sensor networks to a heat conduction process. The performance of the decentralized filters is evaluated and compared to the centralized Kalman filter.

1. Introduction

Current advances in the sensor technology enable the design of small-scale low-cost sensing devices (sensor nodes) endowed with embedded computing and communication capabilities. A number of these sensor nodes can be connected to each other following a certain topology. The collection of such connected sensors is called a sensor network.

There are numerous applications of sensor networks, for instance, in military, health care, or agriculture [1, 11]. An example of the use of sensor networks in the control field is networked control [8]. The use of wireless sensors for feedback has also been reported [5].

An attractive feature of sensor networks is their capability to perform sensing and state estimation in environments with spatially distributed parameters. In this case, sensor nodes are placed at specified locations of the environment to collect measurements that serve as inputs to the filter. To this end, decentralized variants of Kalman filter can be used. Kalman filters form a class of estimators that minimize the variance of the estimation error.

One approach for using the measurements is treating them as one measurement matrix. The measurements from the nodes are then sent to a central processor in which global estimates are computed. In this case the sensor nodes do not have an active role. This approach is called centralized Kalman filter.

As opposed to the centralized approach, the estimate computations can be decentralized to the nodes where local estimates are computed. The local estimates can subsequently be transmitted to a central processor to get global estimates. Alternatively, local estimates are communicated between two or more nodes using an algorithm to get the estimates of the global state. In this case the sensor nodes have an active role to estimate the states.

There are several decentralized Kalman filter methods that have been proposed in the literature. The most representative methods are: parallel information filter [17], distributed information filter [14], distributed Kalman filter with consensus filter [12], and distributed Kalman filter with weighted averaging [3]. This paper compares the mentioned methods to estimate the states of a linear distributed parameter system that has been discretized spatially and temporally resulting in a linear discrete-time large scale system. Note that the above list of the compared methods is not complete and does not include some recent methods, like distributed Kalman filter with diffusion strategies [7]. However, that method closely resembles the distributed Kalman filter with consensus filter [12] and will thus have a similar performance.

A comparison of decentralized Kalman filters has been done in [15], which focused on the communication-related performance. In that paper, the comparison involved the performance of the decentralized Kalman filters with and without data loss. The paper [15] also discussed the required communication for each method and concluded that the performance of the distributed Kalman filter with weighted averaging is the highest with the lowest communication requirement, but that on the other hand, that method also suffers most from data loss. In the current paper, our interest is the performance of the filters related to large-scale linear systems derived from discretization of distributed parameter systems. We evaluate the performance for a system with a high number of states, not only in terms of the evolution of the states over time and space, but also in terms of errors in the steady-state estimation.

The structure of this paper is as follows: after an introduction in Section 1, a brief description of the Kalman filter and its decentralized variants used for our comparison are presented in Section 2. Section 3 introduces a 1D heat conduction process as the benchmark for the comparison and simulations. Section 4 gives conclusions and points out some topics for future work.

2. Kalman filter and its decentralized variants

The Kalman filter is an optimal stochastic discrete-time state estimator developed by Kalman [10]. Since then, the theory and applications of Kalman filters has been treated in different journal papers and books. There are several ways to derive the Kalman filter that can be found in the standard textbooks on optimal estimation, see e.g., [4, 6, 16]. This section presents a brief description of the Kalman filter and its decentralized methods based on [3, 12, 14, 16, 17].

Consider the following process modeled as a discrete-time linear system with measurement equation

$$x(k) = Fx(k-1) + Gu(k-1) + w(k-1) \quad \text{(1a)}$$
$$z(k) = Hx(k) + v(k) \quad \text{(1b)}$$

with $x(k) \in \mathbb{R}^n$ is the state vector, $F \in \mathbb{R}^{n \times n}$ the state matrix, $G \in \mathbb{R}^{n \times m}$ the input matrix, $u \in \mathbb{R}^m$ the input vector, $z(k) \in \mathbb{R}^p$ the measurement vector, $H \in \mathbb{R}^{p \times n}$ the measurement matrix, $w(k)$ and $v(k)$ are the process and measurement noise vector respectively. The process and measurement noise are assumed to be zero-mean Gaussian white noise which have the following properties

$$\mathrm{E}\{w(k)\} = 0 \qquad \mathrm{E}\{w(k)w^{\mathrm{T}}(k)\} = Q$$
$$\mathrm{E}\{v(k)\} = 0 \qquad \mathrm{E}\{v(k)v^{\mathrm{T}}(k)\} = R$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator, the superscript T denotes matrix transpose operation, Q and R are respectively the covariance matrix of the process and measurement noise.

2.1 Centralized Kalman filter
The Kalman filter equations consist of two parts: time update equations and measurement update equations. The following time update equations compute estimates at time step k based on the process model and the previous estimates to get *a priori* estimates:

$$\begin{aligned} \hat{x}^-(k) &= F\hat{x}^+(k-1) + Gu(k-1) \\ P^-(k) &= FP^+(k-1)F^{\mathrm{T}} + Q \end{aligned} \quad \text{(2)}$$

where $P(k)$ is the estimation error covariance matrix, and the superscripts "−" and "+" respectively indicate the *a priori* and *a posteriori* estimates and the error covariance matrix. This step is also referred to as the prediction step. Once the measurements at time step k are available, the

measurement update corrects the *a priori* estimates to get *a posteriori* estimates:

$$\begin{aligned} K(k) &= P^-(k)H^{\mathrm{T}}[HP^-(k)H^{\mathrm{T}} + R]^{-1} \\ \hat{x}^+(k) &= \hat{x}^-(k) + K(k)[z(k) - H\hat{x}^-(k)] \\ P^+(k) &= \left[(P^-(k))^{-1} + H^{\mathrm{T}}R^{-1}H\right]^{-1} \end{aligned} \quad \text{(3)}$$

where $K(k)$ is the Kalman gain matrix. The initial conditions are $\hat{x}^+(0) = x_0$ and $P^+(0) = P_0$, where x_0 and P_0 are respectively the initial guesses of the estimate and estimation error covariance matrix.

The Kalman filter in (2) and (3) is called centralized Kalman filter. It is because the measurements are treated in one measurement matrix. The estimates from the centralized Kalman filter are called global estimates.

Besides the form in (2)–(3), there is another form of the Kalman filter that uses the inverse of the estimation error covariance matrix that is called information matrix, denoted by $\mathcal{I}$ and defined as $\mathcal{I} = P^{-1}$. The filter that use this information matrix is called the information filter. The *a priori* estimate equations of the information filter are equal to (2). The measurement update computations are the following:

$$\begin{aligned} \Delta s(k) &= H^{\mathrm{T}}R^{-1}z(k) \\ \Delta \mathcal{I} &= H^{\mathrm{T}}R^{-1}H \\ \hat{x}^+(k) &= \hat{x}^-(k) + \Delta s(k) \\ \mathcal{I}^+(k) &= \mathcal{I}^-(k) + \Delta \mathcal{I} \end{aligned} \quad \text{(4)}$$

where Δs and $\Delta \mathcal{I}$ are respectively the information vector and matrix update. The information filter avoids the need of matrix inverse computation which is, from a numerical point of view, preferably avoided.

In the application of the decentralized Kalman filter with sensor networks for distributed parameter systems, each node i has a capability to compute its own estimates $\hat{x}_i(k)$ and the corresponding estimation error covariance matrix $P_i(k)$. The estimate and/or the error covariance matrix are communicated to other nodes based on the network topology. In our case, consider a sensor network consisting of N sensor nodes. The nodes are connected to each other, following a specified network topology. In the network, nodes i and j are neighbors if there is a direct link between them. The set of neighbors of node i, and including node i itself is denoted by the set $\mathcal{N}_i$. We assume each node has an identical process model (1a) and the corresponding process noise Q, but a different measurement matrix. Since each node measures one or more state components of the system and since no state component is measured by two or more nodes, we can assume that each local measurement matrix H_i is one block row of the global measurement matrix H. In other words, the global measurement matrix H is the stack of all local measurement matrices H_i

$$H = \begin{bmatrix} H_i \\ \vdots \\ H_N \end{bmatrix} \qquad .$$

So the local measurement in node i is expressed as

$$z_i(k) = H_i x_i(k) + v_i(k) \quad . \tag{5}$$

It is assumed that measurement noise between node i and j is uncorrelated, or $R_{ij} = \mathrm{E}\{v_i(k)v_j^{\mathrm{T}}(k)\} = 0$ for $i \neq j$.

In this paper, measurement updates and the resulting estimates in each node are called local updates and local estimates respectively.

2.2. Parallel information filter (PIF)

The parallel information filter computes local *a posteriori* estimates $\hat{x}_i^+(k)$ and the corresponding estimation error covariance matrix P_i^+ in parallel in each node.

Then $\hat{x}_i^+(k)$ and $P_i^+(k)$ are sent to a central processor in which the estimates are combined to get the global estimate $\hat{x}(k)$ [17]. The time and measurement update equations for node i are

- The local time update:

$$\begin{aligned}
\hat{x}_i^-(k) &= F\hat{x}_i^+(k-1) + Gu(k-1) \\
P_i^-(k) &= FP_i^+(k-1)F^{\mathrm{T}} + Q
\end{aligned} \tag{6}$$

- The local measurement:

$$\begin{aligned}
K_i(k) &= P_i^-(k)H_i^{\mathrm{T}}R_i^{-1} \\
\hat{x}_i^+(k) &= \hat{x}_i^-(k) + K_i(k)[z_i(k) - H_i\hat{x}_i^-(k)] \\
\left(P_i^+(k)\right)^{-1} &= \left(P_i^-(k)\right)^{-1} + H_i^{\mathrm{T}}R_i^{-1}H_i
\end{aligned} \tag{7}$$

In the central processor, estimates from all nodes are combined into one estimate. It is desired that the estimate is as certain as possible, or in other words, an estimate with a low uncertainty is preferable. In case of estimates and uncertainties from N measurements, where the measurement of node i is independent to that of node j for $i \neq j$, estimates with lower uncertainty should be given larger weights. With such consideration, the weight for each measurement can be calculated as [17]

$$\omega_i(k) = \frac{\left(\mathrm{tr}\left\{P_i^+(k)\right\}\right)^{-1}}{\sum_{i=1}^{N}\left(\mathrm{tr}\left\{P_i^+(k)\right\}\right)^{-1}} \tag{8}$$

Once the weights have been determined, the global *a posteriori* estimate and its estimate error covariance matrix can be expressed as follows

$$P^{-1}(k) = \sum_{i=1}^{N} \omega_i(k) \left(P_i^+(k)\right)^{-1} \tag{9a}$$

$$\hat{x}(k) = \sum_{i=1}^{N} \omega_i(k) P(k) \left(P_i^+(k)\right)^{-1} \hat{x}_i^+(k) \tag{9b}$$

This method relies on the central processor to get the global estimates. Hence, it is necessary that all sensor nodes are neighbors of the central processor to assure that all local measurements can be combined into global ones.

2.3. Distributed information filter (DIF)

The decentralized information filter was proposed by Rao and Durrant-Whyte [14] to eliminate the need of a central processor in the decentralized Kalman filter. Using a central processor creates a hierarchy in the network. Furthermore, the network is highly dependent on the central processor. Eliminating the central processor makes that all nodes are at the same level and moreover removes dependency on a single component.

The key idea of this method is expressed in the relation between information vectors and matrix updates, respectively, for the global estimates of the centralized method and local estimates in each node i

$$\begin{aligned}
\Delta s(k) &= H^{\mathrm{T}}R^{-1}z(k) = \sum_{i=1}^{N} H_i^{\mathrm{T}}R_i^{-1}z_i(k) \\
\Delta \mathcal{I} &= H^{\mathrm{T}}R^{-1}H = \sum_{i=1}^{N} H_i^{\mathrm{T}}R_i^{-1}H_i \quad .
\end{aligned} \tag{10}$$

Local updates are computed in each node and sent to the neighboring nodes. Node i adds all information updates from its neighbors to its own updates and then computes the updated estimates and estimation error covariance matrix. Estimates after the communications of the nodes are called communication update estimates.

The time and measurement update equations for node i are

- The local time update equations:

$$\begin{aligned}
\hat{x}_i^-(k) &= F\hat{x}_i^+(k-1) + Gu(k-1) \\
P_i^-(k) &= FP_i^+(k-1)F^{\mathrm{T}} + Q
\end{aligned} \tag{11}$$

- The information vector and matrix update

$$\Delta s_i(k) = H_i^{\mathrm{T}}R_i^{-1}z_i(k), \qquad \Delta \mathcal{I}_i = H_i^{\mathrm{T}}R_i^{-1}H_i$$

- The communication update

$$\begin{aligned}
\hat{x}_i^+(k) &= P_i^+(k)\Bigg[(P_i^-(k))^{-1}\hat{x}_i^-(k) \\
&\qquad + \sum_{j\in\mathcal{N}_i} \Delta s_j(k)\Bigg] \\
(P_i^+(k))^{-1} &= (P_i^-(k))^{-1} + \sum_{j\in\mathcal{N}_i} \Delta \mathcal{I}_j
\end{aligned} \tag{12}$$

This method decentralizes the computations of global estimates to every node without the need of a central processor. If all nodes are fully connected, then (10) shows that the performance of this method is equal to that of the centralized Kalman filter. Note that we include inputs in (11), which is not considered in [14].

2.4. Distributed Kalman filter with consensus filter (DKFCF)

The distributed Kalman filter with consensus filter is proposed by Olfati-Saber [12]. The main feature of this

method is the use of the consensus algorithm to obtain the communication update estimates. The consensus algorithm at node i is performed as follows: for each consensus step, node i receives estimates from its neighbors. Node i subtracts its estimate from the estimate of each of its neighbors, weights the result with factor γ and adds the obtained value to its estimate.

Another feature of this method is availability of stability analysis. Olfati-Saber *et al.* in [13] presented stability analysis of the consensus algorithm using algebraic graph theory.

The time and measurement update equations for node i are

- The local time update equations:

$$\begin{aligned} \hat{x}_i^-(k) &= F\hat{x}_i(k-1) + Gu(k-1) \\ P_i^-(k) &= FP_i^+(k-1)F^{\mathrm{T}} + Q \end{aligned} \qquad (13)$$

- The information vector and matrix update

$$\Delta s_i(k) = H_i^{\mathrm{T}} R_i^{-1} z_i(k), \qquad \Delta \mathcal{I}_i = H_i^{\mathrm{T}} R_i^{-1} H_i$$

- The measurement update

$$\hat{x}_i^+(k) = P_i^+(k)\left[(P_i^-(k))^{-1}\hat{x}_i^-(k) \right. $$
$$\left. + \sum_{j\in\mathcal{N}_i} \Delta s_j(k) \right] \qquad (14)$$
$$(P_i^+(k))^{-1} = (P_i^-(k))^{-1} + \sum_{j\in\mathcal{N}_i} \Delta \mathcal{I}_j$$

- Consensus step, iterated S times

$$\hat{x}_{i,\ell}^+(k) = \hat{x}_{i,\ell-1}^+(k)$$
$$+ \gamma \sum_{j\in\mathcal{N}_i} \left(\hat{x}_{j,\ell-1}^+(k) - \hat{x}_{i,\ell-1}^+(k) \right) \qquad (15)$$

for $\ell = 1,\ldots,S$ where S is the number of iterations.

Up to the consensus step, this method is identical to the distributed information filter.

2.5. Distributed Kalman filter with weighted averaging (DKFWA)

The distributed Kalman filter with weighted averaging has been proposed in [2,3]. A feature of this method is the reduction of computation and communication load. The reduction is because the nodes only compute and send the estimates, without the error estimation covariance matrix.

Different from the previous methods, this method consists of two parts: on-line and off-line. The on-line part computes and communicates estimates. In the off-line part, Kalman gains and weights are computed for each node. In this method, the Kalman gain and the weight W are computed once and used during the entire operation.

The idea of this method is as follows: node i receives estimates from its neighbors and weights them with a weight matrix W. Then the weighted estimates are added to the estimate of node i.

The on-line steps of the distributed Kalman filter with weighted averaging at node i are as follows

- The time update equation

$$\hat{x}_i^{1-}(k) = F\hat{x}_i^{1+}(k-1) + Gu(k-1) \qquad (16)$$

where $\hat{x}_i^1(k)$ denotes the local estimates at node i

- The measurement update equation

$$\hat{x}_i^{1+}(k) = \hat{x}_i^{1-}(k) + K_i[z(k) - H_i\hat{x}_i^{1-}(k)] \qquad (17)$$

- The information exchange equation

$$\hat{x}_i^+(k) = \sum_{j\in\mathcal{N}_i} W_{ij}\hat{x}_j^{1+}(k) \qquad (18)$$

where W_{ij} is the weight of the estimate of node j that is used to compute the global estimates in node i. The value of W_{ij} is zero if node i is not connected to node j.

The off-line computations are performed to minimize the trace of estimation error covariance matrix $P^+(k)$ which is defined as

$$P^+(k) = \mathrm{E}\left\{ (x(k) - \hat{x}^+(k))(x(k) - \hat{x}^+(k))^{\mathrm{T}} \right\}$$

for $x(k) = \begin{bmatrix} x_1(k)^{\mathrm{T}} & \cdots & x_N(k)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and $\hat{x}^+(k) = \begin{bmatrix} \hat{x}_1^+(k)^{\mathrm{T}} & \cdots & \hat{x}_N^+(k)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Using (18) and

$$\sum_{j\in N_i} W_{ij} = I \qquad (19)$$

to get unbiased estimates, we obtain the following relation

$$P^+(k) = W(x(k) - \hat{x}^{1+}(k))(x(k) - \hat{x}^{1+}(k))^{\mathrm{T}}W^{\mathrm{T}}$$
$$= WP^{1+}(k)W^{\mathrm{T}} \qquad (20)$$

The covariance $P^{1+}(k)$ in the last equation can be written as

$$P^{1+}(k) = \begin{pmatrix} I & \tilde{K} \end{pmatrix} \Phi(k) \begin{pmatrix} I & \tilde{K} \end{pmatrix}^{\mathrm{T}} \qquad (21)$$

with

$$\Phi(k) = \begin{pmatrix} I \\ -\tilde{H} \end{pmatrix} P^-(k) \begin{pmatrix} I \\ -\tilde{H} \end{pmatrix}^{T} + \begin{pmatrix} 0 & 0 \\ 0 & R \end{pmatrix}$$

and

$$\tilde{K} = \mathrm{blockdiag}(K_1,\ldots,K_N)$$
$$\tilde{H} = \mathrm{blockdiag}(H_1,\ldots,H_N) \qquad .$$

It should be noted that for this method, it is possible to have a non-diagonal measurement covariance matrix R.

In order to get an optimal filter, it is necessary to find the values of W and $\tilde{K}$ that minimize (21). Instead of direct minimization of (21), the Kalman gain K and weight

Table 1: Characteristics comparison of the different filters

	CKF	PIF	DIF	DKFCF	DKFWA
Central processing	yes	yes	no	no	no
Connectivity	full	full	full	partial	partial
Communication	single	single	single	multi	single
Global estimates	yes	yes	no	no	no

W are computed by solving the following optimization problem:

$$\min_{\tilde{K},W} \text{tr} \left\{ W \left(I \quad \tilde{K} \right) \Phi \left(I \quad \tilde{K} \right)^{\text{T}} W^{\text{T}} \right\} \tag{22}$$

s.t. $W_{ij} = 0$ if node i and j are not connected and (19)

The obtained gain $\tilde{K}$ and weight W are employed in the on-line computation of the states. Details on how to solve the optimization problem (22) are given in [3].

Basically, this method is a consensus filter but with only one information exchange. The motivation for communication limitation is that communication in sensor networks draws more power than computations.

The characteristics comparison of the Kalman filter methods presented in this section is shown in Table 1.

3. The 1D conduction process model and simulations

The Kalman filters presented in the preceding section are simulated to estimate states of heat conduction process of a rod without input, or in other words, the cooling process from an initial temperature. This process is an example of distributed parameter systems that is modeled as a first-order time-derivative, second-order spatial-derivative PDE and specified boundary conditions. Therefore the conduction process has the elements needed to compare different methods of decentralized Kalman filters for distributed parameter systems.

Consider a rod with length L and cross-section radius r. The density, heat capacity, and thermal conductivity of the material are denoted by ρ, C_{p} and κ respectively. Using energy balance equations [9], we can get the following partial differential equation

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C_{\text{p}}} \left[\kappa \frac{\partial^2 T}{\partial x^2} + \frac{h P_{\text{e}}}{A_{\text{T}}} (T_{\text{e}} - T) \right] \tag{23a}$$

$$T(0,t) = T(L,t) = T_{\text{b}} \tag{23b}$$

$$T(x,0) = T_0 \tag{23c}$$

where T is the temperature of the rod, T_{e} the temperature of the environment, h the heat transfer coefficient of surface of the rod, x the spatial coordinate of length, $P_{\text{e}} = 2\pi r$ the perimeter of the rod, and $A_{\text{T}} = \pi r^2$ the area of the longitudinal section. Equations (23b) and (23c) are the boundary conditions and initial condition respectively. The rod's parameters are listed in Table 2.

The simulation of the Kalman filters requires discretization of (23a) in space and time. For the spatial discretization, the central approximation of the second order derivative was employed:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta_x)^2} \tag{24}$$

where Δ_x is the spatial discretization interval. The spatial discretization results in a grid and it is shown at Figure 1. In the figure, the grid index increases from the left to the right. The distance between each grid is Δ_x. Applying (24) to (23a) and simplify the results with respect to the grid point index, (23a) becomes an ordinary partial differential equation as follows

$$\frac{dT_i}{dt} = C_{\text{x}} T_{i-1} - (2C_{\text{x}} + C_{\text{Peh}})T_i + C_{\text{x}} T_{i+1} + C_{\text{Peh}} T_{\text{e}} \tag{25}$$

with i the node index that corresponds to the grid point index and

$$C_{\text{x}} = \frac{\kappa}{\rho C_{\text{p}} \Delta_x^2}; \quad C_{\text{Peh}} = \frac{h P_{\text{e}}}{\rho C_{\text{p}} A_{\text{T}}}$$

For a grid with n grid points on the rod, we have n ODEs from (25), each of which corresponds to a grid point. These ODEs can be expressed as a state space equation $\dot{x} = Ax + Bu$ with $x = \begin{bmatrix} T_1 & \cdots & T_n \end{bmatrix}^{\text{T}}$, $u = \begin{bmatrix} T_{\text{e}} & T_{\text{b}} \end{bmatrix}^{\text{T}}$ at boundaries and $u = T_{\text{e}}$ at the other grid points. The environment temperature T_{e} is $25\,^{\circ}\text{C} = 298.15\,\text{K}$. The state equation is discretized temporally Euler approximation to get a discrete-time linear equation. A number of N sensor nodes are located on the specified grid points and each node i measures the specific temperature T_i.

We will perform two simulations to compare the decentralized Kalman filter methods. The first simulation observes the estimation error performance of the decentralized methods. The goal of the second simulation is to observe the consistency of the decentralized methods. These two simulations are sufficient to assess to performance of the decentralized Kalman filters.
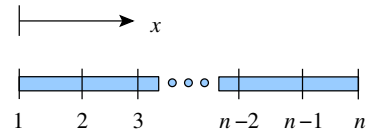

Figure 1: Grid points numbering on the rod

3.1 Simulation 1: estimation error performance

In this simulation, the estimation error performance of the decentralized Kalman filters is compared to the centralized one. The simulation parameters are shown in Table 3. The simulation final time is 5000 s and the sensor node locations are at grid points 2, 104, and 200. In addition,

Table 2: Rod parameters

Parameters	Values	Units
ρ	8700	kg m^{-3}
κ	400	$\text{W m}^{-1}\,\text{K}^{-1}$
C_p	385	$\text{J kg}^{-1}\,\text{K}^{-1}$
h	10	$\text{W m}^{-1}\,\text{K}^{-1}$
T_e	298.15	K

Table 3: Simulation parameters

Parameters	Values	Units
L	4	m
n	201	–
T_0	50	K
x_0	80	–
T_b	298.15	K
σ_v^2	0.04	–
N	3	–

for the DKFCF the number of consensus iterations is 20 and $\gamma = 0.05$. All sensor nodes are fully connected.

The simulation results are shown in Figure 2 where "pt." in the legend stands for "point". The first plot shows the estimates from the CKF in which the true states are plotted in thick grey lines. The other plots, (b)–(e), display the difference between the CKF and the decentralized methods for $\Delta_\text{est} = \hat{x}_\text{ckf} - \hat{x}_\text{d}$. The subscript d in $\hat{x}_\text{d}$ stands for "decentralized". The plotted estimates of the CKF show that the convergence rate of the estimates depend on the spatial distance to the sensor. For instance, estimates of grid point 2 is converge faster than those of grid points 30 and 75. However, the estimates of grid point 2 is noisy due to the measurement noise.

To compare the decentralized methods to the CKF, Figure 2 shows that the performance of the DIF and DKFCF are identical. Furthermore, both methods have the smallest difference to the CKF. If the sensor nodes are fully connected, as in this simulation, the DKFCF is identical to the DIF but with an additional consensus step. The consensus step in the DKFCF costs more communication but does not deliver better estimates due to the full connectivity of the nodes.

The central processor in the PIF basically performs weighted summations to the local estimates based on the estimates' uncertainties. This summing process is not equivalent to the centralized treatment of the measurements in the CKF nor the other more sophisticated methods.

The lower performance of the DKFWA compared to the DIF or DKFCF is expected. The adaptive Kalman gain adapts better to the measurement noise than the fixed gain. Merging with weighted estimates from the neighbors is not enough to fill the gap.

3.2 Simulation 2: steady-state estimation performance

In this simulation, we investigate the steady-state estimates of the filters to assess the consistency of the filters. The simulations are performed by running the simulation from initial temperature until final time 15000 s. The means and variances of the steady-state estimates are computed and plotted in Figure 3. The sensor node locations are at grid 62, 104, and 146. The other simulation parameters are shown also in Table 3.

Figure 3 shows the mean of the estimates in blue circles and the variance of the estimates in blue bars and the mean variance of the measurement in grey. The figure shows that estimates from states that are between two sensors are better. The variance of the estimates for states located before the left-most and after the right-most sensor increase as the distance to the sensor increases. The results show that the spatial distance between the grid point for which the temperature is estimated and the sensor location influences the accuracy of the steady-state estimates.

For the decentralized methods, the DIF and DKFCF give the closest estimates to those of the CKF, while the DKFWA gives the farthest estimates. As already mentioned for Simulation 1, the full connectivity of the nodes results in the equivalence of the CKF and DIF. The result of the PIF is better than the DKFWA. The increase of the estimates' variance of the PIF and the DKFWA is also higher compared to CKF and the other two distributed methods. For the PIF this can be explained as follows: the PIF sums up the estimates, and consequently, their uncertainty. As the uncertainty of estimates increases, the total uncertainty also increases.

4. Conclusion and future work

This paper has presented the comparison of the centralized Kalman filter and some of its variants to a spatially and temporally discretized linear distributed parameter system. In general, the performance of the decentralized Kalman filters cannot be better than the centralized method. The performance of the decentralized methods can be equal to the centralized method provided certain conditions are satisfied. One of them is the full connectivity of the sensor nodes. This situation has been considered in the simulation for the distributed information filter and distributed Kalman filter with consensus filter. The parallel information filter and the distributed Kalman filter with weighted averaging suffer from their approach limitations, e.g., summing up estimates and their uncertainties for the parallel information filter and reducing computation and communication load for the distributed Kalman filter with weighted averaging.

In our comparison, we use linear estimators to a process model that is obtained from a linear partial differential equation. As most systems are intrinsically nonlinear, these linear estimators are not be able to estimate the states of nonlinear systems well. Furthermore, for non-

linear systems, extensions of the Kalman filter have been developed, such as the extended and unscented Kalman filters. Besides filters, there are nonlinear robust observers, e.g., fuzzy robust observers that can also be used in nonlinear and stochastic systems. In our future work, we will compare and assess distributed version of these filters and observers. Comparison of the linear filters for real setups will also be done in the future.

Acknowledgement

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] P. Alriksson and A. Rantzer. Distributed Kalman filtering using weighted averaging. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2006)*, pages 2445–2450, Kyoto, Japan, Jul. 2006.

[3] P. Alriksson and A. Rantzer. Model based information fusion in sensor networks. In *Proceedings of the 17th IFAC World Congress*, pages 4150–4155, Seoul, Korea, Jul. 2008.

[4] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, N.J., 1979.

[5] A. Bemporad, S. Di Cairano, E. Henriksson, and K. H. Johansson. Hybrid model predictive control based on wireless sensor feedback: An experimental study. *International Journal of Robust and Nonlinear Control*, 20(2):209–225, 2010.

[6] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 3rd edition, 1996.

[7] F. Cattivelli and A. Sayed. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084, Sep. 2010.

[8] J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, Jan. 2007.

[9] F. P. Incropera and D. P. Dewitt. *Fundamentals of Heat and Mass Transfer*. John Wiley, 5th edition, 2002.

[10] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[11] M. Liu, N. Patwari, and A. Terzis. Special issue on sensor network applications. *Proceedings of the IEEE*, 98(11):1804–1807, Nov. 2010.
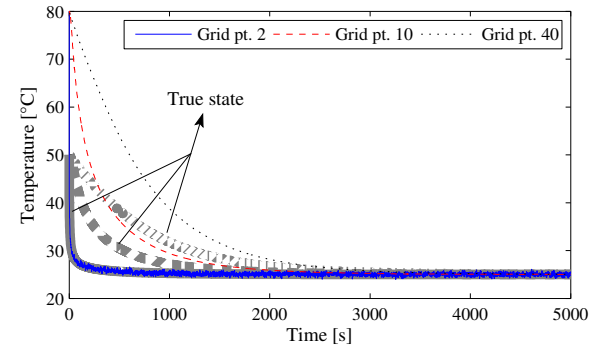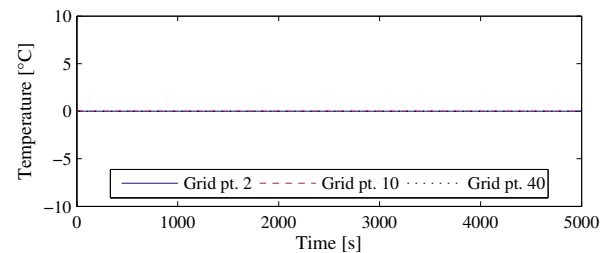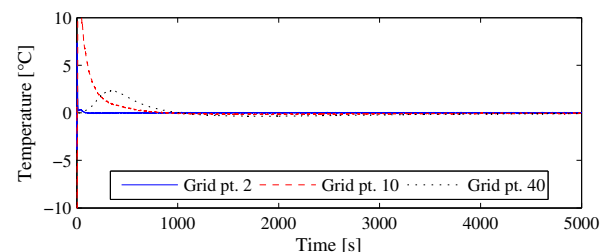

(a) Centralized Kalman filter (CKF)
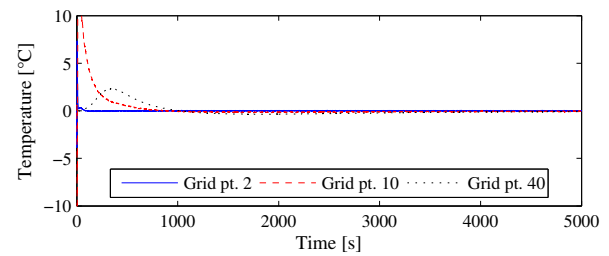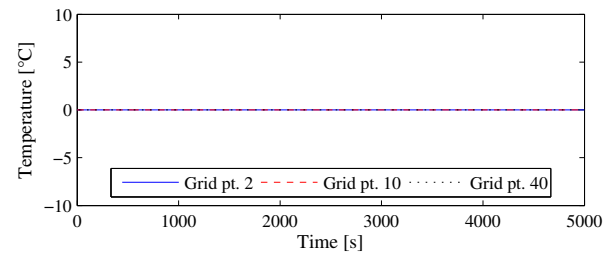
(b) Parallel information filter (PIF)

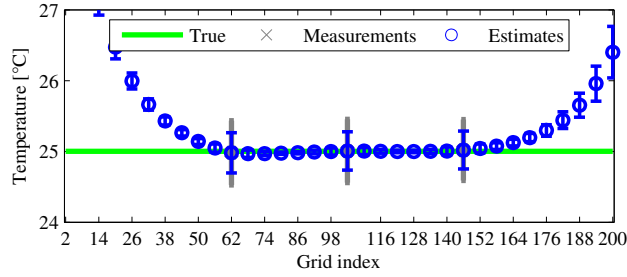(c) Distributed information filter (DIF)
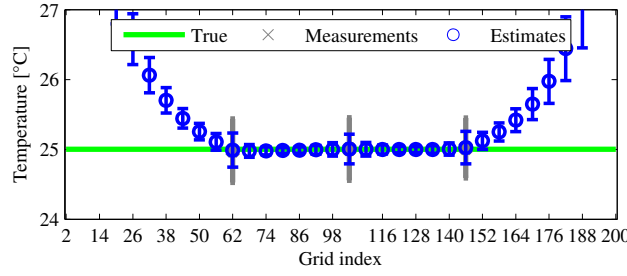

(d) DKF with consensus filter (DKFCF)

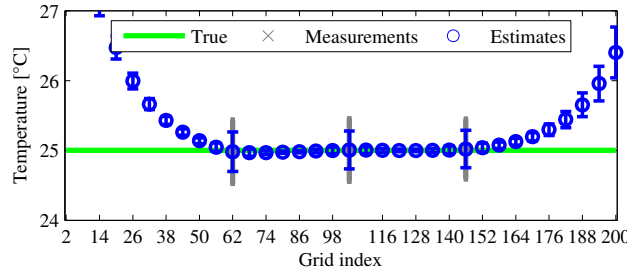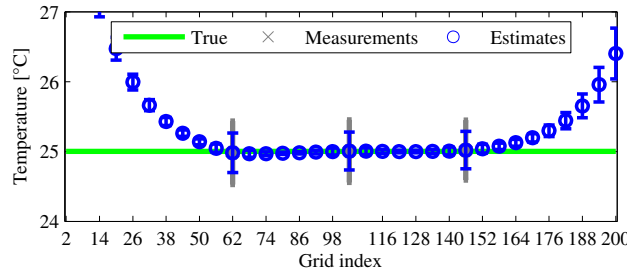(e) DKF with weighted averaging (DKFWA)

Figure 2: Estimation of the decentralized Kalman filters compared to the centralized one
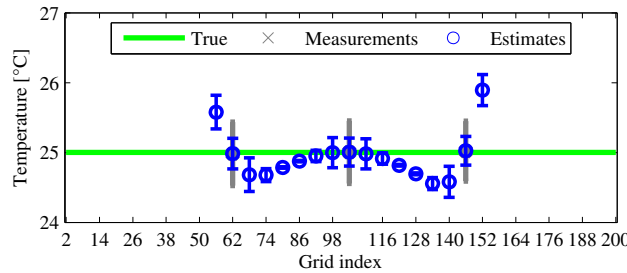
(a) Centralized Kalman filter (CKF)

(b) Parallel information filter (PIF)

(c) Distributed information filter (DIF)

(d) DKF with consensus filter (DKFCF)

(e) DKF with weighted averaging (DKFWA)

Figure 3: Steady state estimation of the centralized and decentralized Kalman filters

[12] R. Olfati-Saber. Distributed Kalman filtering and sensor fusion in sensor networks. In P. J. Antsaklis and P. Tabuada, editors, *Networked Embedded Sensing and Control*, number 331 in Lecture Notes in Control and Information Sciences, pages 157–167. Springer Berlin/Heidelberg, Germany, Jul. 2006.

[13] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan. 2007.

[14] B. S. Rao and H. F. Durrant-Whyte. Fully decentralised algorithm for multisensor Kalman filtering. *IEE Proceedings D: Control Theory and Applications*, 138(5):413–420, Sep. 1991.

[15] J. Sijs, M. Lazar, P. van den Bosch, and Z. Papp. An overview of non-centralized Kalman filters. In *Proceedings of IEEE International Conference on Control Applications (CCA 2008)*, pages 739–744, San Antonio, TX, USA, Sep. 2008.

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

[17] J. L. Speyer. Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem. *IEEE Transactions on Automatic Control*, 24(2):266–269, Apr. 1979.