

Nonlinear estimation with a network of heterogeneous estimators

Citation for published version (APA):

Sijs, J., Papp, Z., & Booij, P. (2011). Nonlinear estimation with a network of heterogeneous estimators. In *Proceedings of the 8th IEEE International Conference on Networking, Sensing and Control (ICNSC '11)*, 11-13 April 2011, Delft, The Netherlands (pp. 433-438). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ICNSC.2011.5874909>

DOI:

[10.1109/ICNSC.2011.5874909](https://doi.org/10.1109/ICNSC.2011.5874909)

Document status and date:

Published: 01/01/2011

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.

Nonlinear estimation with a network of heterogenous algorithms

J. Sijs , Z. Papp and P. Booij.

Abstract—Centralized state-estimation algorithms, such as the original Kalman filter, are no longer feasible in large scale sensor networks, due to practical limitations on communication bandwidth and spatial distribution of resources. To cope with these limitations, various distributed estimation algorithms have been proposed that estimate the state of a process in each sensor node using local measurements. State fusion of this local estimate with the estimates obtained in neighboring nodes ensures that the difference between local estimates is reduced. A common perspective in distributed state-estimation is that each individual node performs the same algorithm locally. This paper investigates whether it is beneficial to have some nodes that can perform a different, more accurate estimation method, i.e., heterogenous. To that extent, a networked system where each node employs the same local state-estimator is compared to a similar system where different nodes can perform different types of local estimation algorithms. Their performances are assessed on a Van-der-Pol oscillator and on a benchmark application to estimate speed profiles in traffic shockwaves. The results of these examples encourage further investigation of heterogeneous, distributed state-estimation.

Index Terms—Nonlinear, distributed, state estimation.

I. INTRODUCTION

Some well known state-estimators for a process with Gaussian noise distributions are the Kalman filter (KF), the extended Kalman filter (EKF) and the unscented Kalman filter (UKF), presented in [1]–[3], respectively. Their centralized algorithms estimate the full state-vector based on a complete set of measurements on the process. Nowadays, measurements are often acquired by means of a sensor network, especially in large-scale spatially distributed processes, e.g., [4]. Employing a centralized state-estimator requires global communication and central data-processing. Since this is known to be infeasible for large-scale sensor networks, the centralized algorithm is distributed among the nodes in the network. Each node then performs an “reduced” estimation algorithm locally, by which a local estimate of the full state-vector is calculated. This local estimate is based on local measurements together with data from neighboring nodes (the “neighboring node” term represents any subset of nodes which is reachable via communication and it is not necessarily the physical proximity), as depicted in Figure 1. The communication and computational requirements of a single node, in these distributed state-estimation set-ups (DSE), scale with the number of neighboring nodes rather than the total network size.

The objective of DSE is to diminish the difference between all local estimates in the network. This, while nodes are only allowed to receive and process the data of their neighboring nodes. To attain this characterization, each node

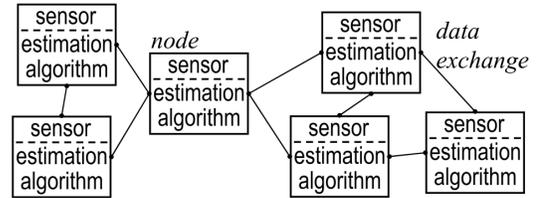


Fig. 1. Distributed state-estimation in a sensor network, where each node performs a “reduced” estimation algorithm locally.

performs a 2-step algorithm consisting of a local state-estimator (LSE) and a local state-fusion method (LSF); 1. the LSE estimates a local version of the full state-vector using the local measurement and 2. the LSF fuses the local estimate with the estimates that are received from neighboring nodes. An additional property, due to the second “fusion” step, is that the covariance of a local state-estimate depends on all information that is available in the network. This aspect of the DSE will be denoted as the *global covariance* property.

Commonly, the LSE of every node is derived from the same type of (centralized) state-estimator, e.g. [5]–[9]. The main contribution of this research is to design and analyze a network of state-estimators, where different nodes perform different types of LSEs. Such a heterogeneous DSE allows different computational requirements per node in the network and thus enhances feasibility of DSE in sensor networks. Also, nodes that are added to an existing network can use arbitrary LSE, while still exchanging estimates with neighboring nodes for state-fusion. Incorporation of these new nodes in an existing sensor network is guaranteed from the *global covariance* property, without the need of reprogramming existing nodes. A case study of the nonlinear Van-der-Pol oscillator shows that inaccurate estimates of nodes that perform the KF-algorithm are improved when a few nodes in the network employ the EKF or UKF as their LSE. Similar results are also shown in a benchmark application when monitoring traffic shockwaves on highways.

II. PRELIMINARIES

\mathbb{R} , \mathbb{R}_+ , \mathbb{Z} and \mathbb{Z}_+ define the set of real numbers, non-negative real numbers, integer numbers and non-negative integer numbers, respectively. Let $\mathbb{X} \subset \mathbb{R}$ be given, then $\mathbb{Z}_{\mathbb{X}} := \mathbb{Z} \cap \mathbb{X}$. In case $\mathbb{X} = \{x_1, \dots, x_m\} \subset \mathbb{R}^n$, where $x_q \in \mathbb{R}^n$ for all $q \in \mathbb{Z}_{[1,m]}$, then $[\mathbb{X}]_q := x_q$. The transpose and inverse of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted as A^T and A^{-1} , respectively. Furthermore, $[A]_{qr} \in \mathbb{R}$ denotes the element in the q -th row and r -th column of A , whereas $[A]_{:q} \in \mathbb{R}^n$ denotes the entire q -th column of A . Given a square matrix $A \in \mathbb{R}^{n \times n}$, let $\lambda_q(A)$ denote the q^{th} eigenvalue of A and let

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

$A = SDS^{-1}$ represent the Jordan decomposition of A , for some $S \in \mathbb{R}^{n \times n}$ and $D := \text{diag}(\lambda_1(A), \lambda_2(A), \dots, \lambda_n(A))$.

Let the function $f(x, y) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^l$ of $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ be given. Then the Jacobian of $f(x, y)$ towards x and towards y is denoted with $\nabla_x f$ and $\nabla_y f$, respectively. Moreover, $\nabla_x f(a, b)$ denotes the value of $\nabla_x f$ in case $x = a$ and $y = b$. The Gaussian function (shortly noted as Gaussian) is denoted as $G(x, \mu, P)$, for some $x, \mu \in \mathbb{R}^n$ and $P \in \mathbb{R}^{n \times n}$. If $G(x, \mu, P)$ is a probability density function (PDF) of the random vector x , then by definition the mean and covariance-matrix of x are μ and P , respectively.

III. PROBLEM FORMULATION

Let us assume an autonomous, nonlinear process that is observed by a sensor network. The state-vector of this process, denoted as $x \in \mathbb{R}^n$, is affected by process noise, which is denoted as $w \in \mathbb{R}^m$. The neighbors of a node i are collected in the set \mathcal{N}_i . Each node i performs a measurement $y_i \in \mathbb{R}^{l_i}$ that is affected by measurement noise, denoted as $v_i \in \mathbb{R}^{l_i}$. The discrete-time, nonlinear process model, given $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}^{l_i}$ for any node i , is described as follows,

$$x(k) = f(x(k-1), w(k-1)), \quad (1a)$$

$$y_i(k) = g_i(x(k)) + v_i(k). \quad (1b)$$

Both the process noise and the measurement noise are assumed to have a zero-mean Gaussian PDF for all k , i.e.,

$$p(w(k)) := G(w(k), 0, W) \text{ and } p(v_i(k)) := G(v_i(k), 0, V_i).$$

The sensor network aims to estimate x in every node by means of a DSE. To that extent, each node i performs an estimation algorithm, of which a schematic set-up is depicted in Figure 2. In line with current literature, each node i employs a ‘‘local state-estimator’’ (LSE) given y_i . The resulting estimate of this LSE at node i at sample instant k is described with the Gaussian PDF $p_i(x(k)) = G(x(k), \hat{x}_i(k), P_i(k))$, for some $\hat{x}_i(k) \in \mathbb{R}^n$ and $P_i(k) \in \mathbb{R}^{n \times n}$. To diminish the difference in local estimates, nodes exchange this PDF with neighboring nodes. As such, node i receives $p_j(x(k))$ for all $j \in \mathcal{N}_i$. The received PDFs are then merged with $p_i(x(k))$ in a ‘‘local state-fusion’’ algorithm (LSF), which results in a fused PDF that is denoted by $p_{i_f}(x(k)) = G(x(k), \hat{x}_{i_f}(k), P_{i_f}(k))$, for some $\hat{x}_{i_f}(k) \in \mathbb{R}^n$ and $P_{i_f}(k) \in \mathbb{R}^{n \times n}$.

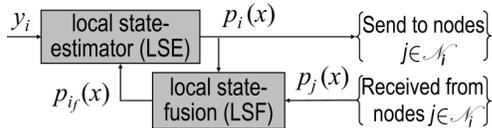


Fig. 2. Schematic set-up of the local estimation algorithm at node i .

To complete the algorithm of a node, two aspects are to be addressed. One is fusion of state-estimates that are described by a Gaussian PDF. For clarity, the fusion method is regarded as a problem of merging any two different estimates of the same state-vector x , i.e., $p_i(x)$ and $p_j(x)$. Since keeping track of shared estimates between different nodes is

intractable, this fusion method cannot require knowledge on the correlation of $p_i(x)$ and $p_j(x)$. The second aspect is a complete description of a node’s algorithm, as it is depicted in Figure 2, with a focus on three different LSE methods, i.e., KF, EKF and UKF. Let us continue with a choice and motivation of the state-fusion method.

IV. STATE FUSION: ELLIPSOIDAL INTERSECTION

This section summarizes a recently developed state-fusion method ‘‘Ellipsoidal intersection’’, as presented in [10]. The method fuses $p_i(x) := G(x, \hat{x}_i, P_i)$ with $p_j(x) := G(x, \hat{x}_j, P_j)$ into the new estimate $p_{i_f}(x) := G(x, \hat{x}_{i_f}, P_{i_f})$, for some $\hat{x}_i, \hat{x}_j, \hat{x}_{i_f} \in \mathbb{R}^n$ and $P_i, P_j, P_{i_f} \in \mathbb{R}^{n \times n}$. It was already shown in [10] that employing this fusion method as LSF, and a KF as LSE, results in a DSE with the *global covariance* property. Alternative fusion methods are found in [11]–[13]. The main reason for choosing Ellipsoidal intersection is its distinguished performance with respect to accuracy in combination with the required computational power.

The first stage of Ellipsoidal intersection is a parametrization of the correlation between $p_i(x)$ and $p_j(x)$. This is done by introducing a new estimate that is based on *mutual* data of $p_i(x)$ and $p_j(x)$. Mutual implies that the same measurements or models were used in both $p_i(x)$ and $p_j(x)$. Similarly, *exclusive* data refers to, for example, measurements that were used in either $p_i(x)$ or $p_j(x)$. This mutual estimate is denoted with $p_\gamma(x) = G(x, \gamma, \Gamma)$, for some ‘‘mutual mean’’ $\gamma \in \mathbb{R}^n$ and ‘‘mutual covariance’’ $\Gamma \in \mathbb{R}^{n \times n}$. On the assumption that $p_i(x)$ and $p_j(x)$ are uncorrelated, i.e., no mutual data, then [5] proves that $p_{i_f}(x)$ is characterized by $P_{i_f}^{-1} = P_i^{-1} + P_j^{-1}$ and $\hat{x}_{i_f} = P_{i_f}(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j)$. In case $p_i(x)$ and $p_j(x)$ are correlated, and the values for γ and Γ are known, then the fused mean \hat{x}_{i_f} and fused covariance P_{i_f} become

$$P_{i_f} = \left(P_i^{-1} + P_j^{-1} - \Gamma^{-1} \right)^{-1}, \quad (2)$$

$$\hat{x}_{i_f} = P_{i_f} \left(P_i^{-1} \hat{x}_i + P_j^{-1} \hat{x}_j - \Gamma^{-1} \gamma \right).$$

The second stage is determining γ and Γ when correlation is unknown. Therefore, to ensure that $p_i(x)$ is updated with exclusive information of $p_j(x)$, values for Γ and γ are derived by assuming a maximum effect of mutual information. See [10] for more details. To that extent, the matrices S_i , D_i , S_j and D_j are introduced via the Jordan decompositions

$$P_i = S_i D_i S_i^{-1} \text{ and } D_i^{-0.5} S_i^{-1} P_j S_i D_i^{-0.5} = S_j D_j S_j^{-1}.$$

Also, let $H := P_i^{-1} + P_j^{-1} - 2\Gamma$ and let $\lambda_{0+}(H) \in \mathbb{R}_+$ denote the smallest, non-zero eigenvalue of H . Then the mutual covariance and the mutual mean according to [10], for some $\eta, c \in \mathbb{R}_+$, are given as follows

$$\Gamma = S_i D_i^{0.5} S_j D_j S_j^{-1} D_i^{0.5} S_i^{-1}, \quad (3)$$

$$\gamma = \left(P_i^{-1} + P_j^{-1} - 2\Gamma^{-1} + 2\eta I \right)^{-1} \times \left(\left(P_j^{-1} - \Gamma^{-1} + \eta I \right) \hat{x}_i + \left(P_i^{-1} - \Gamma^{-1} + \eta I \right) \hat{x}_j \right). \quad (4)$$

Where,

$$[D_\Gamma]_{qr} = \begin{cases} \max([D_j]_{qr}, 1) & \text{if } q = r, \\ 0 & \text{if } q \neq r, \end{cases}$$

and $\eta = \begin{cases} 0 & \text{if } |H| \neq 0, \\ c \ll \lambda_{0+}(H) & \text{if } |H| = 0. \end{cases}$

Ellipsoidal intersection of (2) is employed as LSF. Further aspects on realization of this method in the combined algorithm of LSE and LSF is presented next.

V. A HETEROGENEOUS, DISTRIBUTED STATE-ESTIMATOR

Algorithm V.1 is a detailed description of the set-up as it is depicted in Figure 2. Therein, “LocalStateEst” denotes the algorithm of the LSE, i.e., KF, EKF and UKF, which are presented in more detail after Algorithm V.1. Notice that $\hat{x}_i(k)$ and $P_i(k)$ of this LSE are based on fused estimates at $k-1$, i.e., $\hat{x}_{i_f}(k-1)$ and $P_{i_f}(k-1)$. Fusion of one estimate with multiple other estimates is commonly conducted recursively. This means that the LSF algorithm fuses $p_i(x(k))$ with the first received $p_j(x(k))$, after which their resulting fused estimate is further merged with the PDF that is received next, and so on. Let the initial local estimate at sample-instant k be denoted as $p_{i(0)}(x) := p_i(x(k))$. Then this recursive behavior implies that $p_{i(l)}(x)$, for all $l \in \mathbb{Z}_{[1,L]}$ and $L := \#\mathcal{N}_{i(1)}$, is defined as the fused estimate of $p_{i(l-1)}(x)$ and the l -th received estimate $p_j(x(k))$, which will be denoted as $p_{j(l)}(x)$. The final estimate after fusing $p_i(x(k))$ with all received PDFs is thus $p_{i_f}(x(k)) := p_{i(L)}(x)$. Hence, each node i performs Algorithm V.1 at each sample-instant k , i.e.,

Algorithm V.1 Heterogeneous DSE (HDSE)

$$(\hat{x}_i(k), P_i(k)) = \text{LocalStateEst}(\hat{x}_{i_f}(k-1), P_{i_f}(k-1), y_i(k));$$

$$\hat{x}_{i(0)} = \hat{x}_i(k), \quad P_{i(0)} = P_i(k);$$

for $l = 1, \dots, L$, do:

$$\hat{x}_{j(l)} = \hat{x}_j(k), \quad P_{j(l)} = P_j(k), \quad j \in \mathcal{N}_i;$$

$$\Gamma_l = \text{MutualCovariance}(P_{i(l-1)}, P_{j(l)}), \quad (3);$$

$$\gamma_l = \text{MutualMean}(P_{i(l-1)}, P_{j(l)}, \Gamma, \hat{x}_{i(l-1)}, \hat{x}_{j(l)}), \quad (4);$$

$$P_{i(l)} = \left(P_{i(l-1)}^{-1} + P_{j(l)}^{-1} - \Gamma_l^{-1} \right)^{-1};$$

$$\hat{x}_{i(l)} = P_{i(l)} \left(P_{i(l-1)}^{-1} \hat{x}_{i(l-1)} + P_{j(l)}^{-1} \hat{x}_{j(l)} - \Gamma_l^{-1} \gamma_l \right);$$

end

$$\hat{x}_{i_f}(k) = \hat{x}_{i(L)}, \quad P_{i_f}(k) = P_{i(L)}; \quad \square$$

A. Kalman filter

In general, employing a KF as LSE results in a high estimation error and low computational power. This is mainly due to a linear approximation of the nonlinear process-model of (1). A description of this approximated model, for some $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C_i \in \mathbb{R}^{l_i \times n}$, yields,

$$x(k) = Ax(k-1) + Bw(k-1), \quad y_i(k) = C_i x(k) + v_i(k). \quad (5)$$

Let $\hat{x}_i(k^-) \in \mathbb{R}^n$ and $P_i(k^-) \in \mathbb{R}^{n \times n}$ denote the predicted mean and covariance at node i at sample instant k , respectively.

Then the KF algorithm computes $\hat{x}_i(k)$ and $P_i(k)$ as follows,

$$\begin{aligned} \hat{x}_i(k^-) &= A\hat{x}_{i_f}(k-1), \\ P_i(k^-) &= AP_{i_f}(k-1)A^\top + BWB^\top, \\ K_i(k) &= P_i(k^-)C_i^\top \left(C_i P_i(k^-)C_i^\top + V_i \right)^{-1}, \\ \hat{x}_i(k) &= \hat{x}_i(k^-) + K_i(k) \left(y_i(k) - C_i \hat{x}_i(k^-) \right), \\ P_i(k) &= (I - K_i(k)C_i) P_i(k^-). \end{aligned} \quad (6)$$

B. Extended Kalman filter

In case an EKF is employed as LSE, then the nonlinear model of (1) is used to calculate a predicted mean, i.e., $\hat{x}_i(k^-)$. To predict the covariance, i.e. $P_i(k^-)$, the nonlinear dynamics of $x(k-1)$ are linearized around its current working point $\hat{x}_i(k^-)$. Linearizing these dynamics is obtained via Jacobian matrices for both nonlinear functions of (1), i.e., $F_i(k) := \nabla_x f(\hat{x}_i(k-1), 0)$, $E_i(k) := \nabla_w f(\hat{x}_i(k-1), 0)$ and $H_i(k) := \nabla_x g_i(\hat{x}_i(k-1))$. The predicted state-estimates and Jacobian matrices are then used to calculate $\hat{x}_i(k)$ and $P_i(k)$ similar to (6), i.e.,

$$\begin{aligned} \hat{x}_i(k^-) &= f(\hat{x}_{i_f}(k-1), 0), \\ P_i(k^-) &= F_i(k)P_{i_f}(k-1)F_i^\top(k) + E_i(k)WE_i^\top(k), \\ K_i(k) &= P_i(k^-)H_i^\top(k) \left(H_i(k)P_i(k^-)H_i^\top(k) + V_i \right)^{-1}, \\ \hat{x}_i(k) &= \hat{x}_i(k^-) + K_i(k) \left(y_i(k) - g_i(\hat{x}_i(k^-)) \right), \\ P_i(k) &= (I - K_i(k)H_i(k))P_i(k^-). \end{aligned} \quad (7)$$

Although an EKF enjoys low computational power, its accuracy depends on the support to linearize the process-model.

C. Unscented Kalman filter

In case an UKF is employed as LSE, then the nonlinear model of (1) is applied to various state values of $x(k-1)$ and noise values of $w(k-1)$. These values are selected from an augmented vector space, for which this augmented vector $\mu \in \mathbb{R}^{n+m}$ combines the state and process noise, i.e., $\mu := \begin{pmatrix} x \\ w \end{pmatrix}$. Since x and w are defined with a Gaussian PDF, also μ is described with a Gaussian PDF having a mean $\hat{\mu}_i \in \mathbb{R}^{n+m}$ and covariance $U_i \in \mathbb{R}^{(n+m) \times (n+m)}$. Their values at a sample-instant $k-1$ follow from $p_{i_f}(x(k-1))$ and $p(w(k-1))$, i.e.,

$$\hat{\mu}_i(k-1) := \begin{pmatrix} \hat{x}_{i_f}(k-1) \\ 0 \end{pmatrix}, \quad U_i(k-1) := \begin{pmatrix} P_{i_f}(k-1) & 0 \\ 0 & W \end{pmatrix}.$$

This mean and covariance are then used to select $M := 2(n+m) + 1$ different values of $\mu(k-1)$. The collection of all these selected vectors is denoted with the set $\mathbb{U}(k-1) \subset \mathbb{R}^{n+m}$, i.e., $[\mathbb{U}(k-1)]_q \in \mathbb{R}^{n+m}$ denotes the q -th selected value of $\mu(k-1)$. This value of $[\mathbb{U}(k-1)]_q$, for all $q \in \mathbb{Z}_{[1,M]}$ and some $\tilde{\mu}_q \in \mathbb{R}^{n+m}$, $c \in \mathbb{R}_+$, is defined as follows:

$$[\mathbb{U}(k-1)]_q := \hat{\mu}_i(k-1) + c\tilde{\mu}_q(q), \quad \text{where}$$

$$\tilde{\mu}_q(q) := \begin{cases} [U_i^{0.5}(k-1)]_{:q} & \text{if } q \in \mathbb{Z}_{[1,n+m]}, \\ -[U_i^{0.5}(k-1)]_{:(q-n-m)} & \text{if } q \in \mathbb{Z}_{[n+m+1,M-1]}, \\ 0 & \text{if } q = M. \end{cases}$$

The process-model of (1) is applied on each selected vector $[\mathbb{U}(k-1)]_q$ and results in a prediction of $x(k)$. Therefore,

let $\mathbb{X}(k^-) \subset \mathbb{R}^n$ denote a set of predicted state-vectors and $\mathbb{Y}(k^-) \subset \mathbb{R}^{l_i}$ a set of predicted measurements. Then the UKF defines each prediction as follows:

$$[\mathbb{X}(k^-)]_q := f([\mathbb{U}(k-1)]_q), \quad [\mathbb{Y}(k^-)]_q := g_i([\mathbb{X}(k^-)]_q).$$

The set $\mathbb{X}(k^-)$ is used to calculate a predicted state-mean $\hat{x}_i(k^-)$ and state-covariance $P_i(k^-)$. Similarly, $\mathbb{Y}(k^-)$ is used to determine a mean and covariance of the predicted measurement, which are denoted as $\hat{y}_i(k^-) \in \mathbb{R}^{l_i}$ and $R_i(k^-) \in \mathbb{R}^{l_i \times l_i}$, respectively. Values of these predicted variables, for some weights $\omega_q \in \mathbb{R}_+$ and for all $q \in \mathbb{Z}_{[1,M]}$, are defined with the following convex combinations:

$$\hat{x}_i(k^-) = \sum_{q=1}^M \omega_q [\mathbb{X}(k^-)]_q, \quad (8a)$$

$$P_i(k^-) = \sum_{q=1}^M \omega_q ([\mathbb{X}(k^-)]_q - \hat{x}_i(k^-)) ([\mathbb{X}(k^-)]_q - \hat{x}_i(k^-))^\top,$$

$$\hat{y}_i(k^-) = \sum_{q=1}^M \omega_q [\mathbb{Y}(k^-)]_q, \quad (8b)$$

$$R_i(k^-) = \sum_{q=1}^M \omega_q ([\mathbb{Y}(k^-)]_q - \hat{y}_i(k^-)) ([\mathbb{Y}(k^-)]_q - \hat{y}_i(k^-))^\top.$$

Common values for the constant c and the weights ω_q are:

$$c = \sqrt{n+m}, \quad \omega_M = 0, \quad \omega_q = \frac{1}{2(n+m)}, \quad \forall q \in \mathbb{Z}_{[1,M-1]}.$$

Finally, $\hat{x}_i(k)$ and $P_i(k)$ are calculated by comparing $\hat{y}_i(k^-)$ with the measured value $y_i(k)$. In their expression, $\hat{x}_i(k)$ and $P_i(k)$ make use of the cross-covariance of $\mathbb{X}(k^-)$ and $\mathbb{Y}(k^-)$, which is denoted as $S_i(k^-) \in \mathbb{R}^{n \times l_i}$ and is determined similarly as to $P_i(k^-)$ and $R_i(k^-)$, i.e.,

$$S_i(k^-) = \sum_{q=1}^M \omega_q ([\mathbb{X}(k^-)]_q - \hat{x}_i(k^-)) ([\mathbb{Y}(k^-)]_q - \hat{y}_i(k^-))^\top,$$

$$\hat{x}_i(k) = \hat{x}_i(k^-) + S_i(k^-) (R_i(k^-) + V_i)^{-1} (y_i(k) - \hat{y}_i(k^-)),$$

$$P_i(k) = P_i(k^-) - S_i(k^-) (R_i(k^-) + V_i) S_i(k^-).$$

Employing an UKF results in a low estimation error, at the costs of high computational power. The performance of the KF, EKF and UKF in a heterogenous set-up of the DSE is analyzed next.

VI. A CASE STUDY: THE VAN-DER-POL OSCILLATOR

In this section the heterogenous, distributed state-estimator (HDSE) is analyzed on its estimation error. The goal is estimating the two states of a Van-der-Pol oscillator, i.e., $[x(k)]_1$ and $[x(k)]_2$. The process is measured by 5 sensor-nodes and each node can only communicate with its direct neighbors, as depicted in Figure 3.

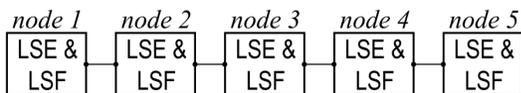


Fig. 3. Network set-up of the case study of the Van-der-Pol oscillator.

The discrete-time process-model, with $\delta \in \mathbb{R}_+$ defined as the sampling time, yields,

$$[x(k)]_1 = [x(k-1)]_1 + \delta [x(k-1)]_2 + [w(k-1)]_1,$$

$$[x(k)]_2 = (1 + 0.5\delta) [x(k-1)]_2 + f_2(x(k-1)) + [w(k-1)]_2,$$

where,

$$f_2(x(k-1)) := \delta [x(k-1)]_1 (0.5 [x(k-1)]_1 [x(k-1)]_2 - 1).$$

Let $x(0) = \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}$, after which both state-elements will start to oscillate around 0 with a amplitude of 2. Since $[x]_1$ and $[x]_2$ are both sinusoids and have a difference in phase-shift of $\frac{\pi}{2}$, one can approximate that $0.5\delta([x(k-1)]_1)^2[x(k-1)]_2$ is average by 0. Hence, a linearized model of the Van-der-Pol Oscillator according to the description of (5), yields,

$$A = \begin{pmatrix} 1 & \delta \\ -\delta & 1 + 0.5\delta \end{pmatrix} \text{ and } B = I.$$

For each measurement y_i at node i let us define the following measurement-matrices: $C_1 = (1 \ 0)$, $C_2 = (1 \ 0)$, $C_3 = (0 \ 1)$, $C_4 = (0 \ 1)$ and $C_5 = (1 \ 0)$. Their corresponding measurement noise v_i is characterized by $R_1 = 0.8$, $R_2 = 1$, $R_3 = 0.8$, $R_4 = 1$ and $R_5 = 1.5$. In this set-up each node either employs a KF or UKF as LSE and $\delta = 0.1$ seconds. All LSEs in the network start with the same initial state-estimates, i.e., $\hat{x}_i(0) = \begin{pmatrix} 2 \\ -0.3 \end{pmatrix}$ and $P_i(0) = 5I$, for all $i \in \mathbb{Z}_{[1,5]}$. The process-noise $w(k)$ is such that $\text{cov}(w(k)) = 10^{-3}I$, for all $k \in \mathbb{Z}_+$. Therefore, $W = 10^{-3}I$ for the UKF. The linearized model, as it is used by a KF, results in reduced accuracy since it does not take $0.5\delta([x(k-1)]_1)^2[x(k-1)]_2$ of f_2 into account. This inaccuracy is modeled via increased process noise for the KF, i.e., $W = 0.5\delta \text{cov}([x(k-1)]_1)^2[x(k-1)]_2 I \approx 10^{-1}I$.

In this simulation three different DSEs are compared that all perform Algorithm V.1 in each node. In the first DSE all nodes employ an UKF as their LSE and is therefore denoted with DUKF. In the second DSE all nodes employ a KF as their LSE and is therefore denoted with DKF. The third DSE performs the HDSE, where nodes 1, 2, 4 and 5 perform a KF-algorithm as their LSE and node 3 employs a UKF. The three DSEs are compared in Figure 4 and Figure 5 on their resulting squared estimation errors of the individual state-elements at node 2 and 5, respectively.

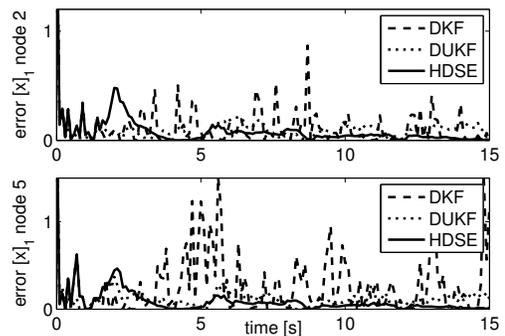


Fig. 4. Squared estimation error of the first state element, i.e., $([\hat{x}_i(k)]_1 - [x(k)]_1)^2$, at node 2 and 5.

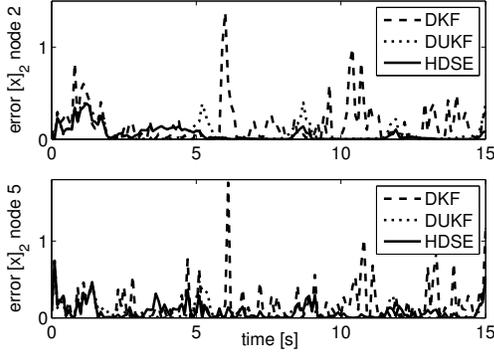


Fig. 5. Squared estimation error of the second state element, i.e., $([\hat{x}_i(k)]_2 - [x(k)]_2)^2$, at node 2 and 5.

	μ_1	Σ_1	μ_3	Σ_3	μ_4	Σ_4
DKF	0.40	0.14	0.33	0.10	0.51	0.31
DUKF	0.12	0.02	0.12	0.03	0.13	0.03
HDSE	0.15	0.04	0.12	0.03	0.12	0.03

TABLE I

THE MEAN μ_i AND THE COVARIANCE Σ_i OF THE ESTIMATION ERROR $(x(k) - \hat{x}_i(k))^T (x(k) - \hat{x}_i(k))$, FOR NODES 1, 3 AND 4.

Figures 4 and 5 show that the estimation-error of the HDSE in node 2 and 5 is reduced compared to the DKF. Notice that these DSEs only differ in node 3, where the HDSE employs a UKF as its LSE rather than a KF. This shows that substituting one KF in the network with a more accurate LSE improves estimation at all other nodes as well. A similar analysis of the other nodes is presented in Table I. Therein, the mean and covariance of the squared estimation-error at nodes 1, 3 and 4. This table indicates that also at these nodes the HDSE has a smaller estimation error than the DKF and has similar results as the DUKF.

It was shown that replacing a KF of a node with the more accurate UKF improves local estimates at all nodes in the network. Next, let us show that adding nodes to an existing network is also beneficial, even when the added nodes perform a LSE-algorithm that is less accurate than current LSEs in the network. To that extent, node 3 of the HDSE is compared to a node which only performs an UKF using y_3 , denoted as IUKF. Notice that this IUKF is a result of the HDSE at node 3 when no communication is allowed. Figure 6 shows the resulting squared estimation-error of both estimators in node 3.

Figure 6 shows that adding nodes to an existing network can improve estimation results, even if additional nodes are not as accurate as the initial set of nodes. Nonetheless, such improvement only holds when the process and measurement noise have realistic values for their corresponding PDF. This means that nodes which employ a KF should account for linearization of the process model, for example, by increasing their covariance of the process noise. Next, the case study is extended to a benchmark application of estimating traffic shockwaves on highways.

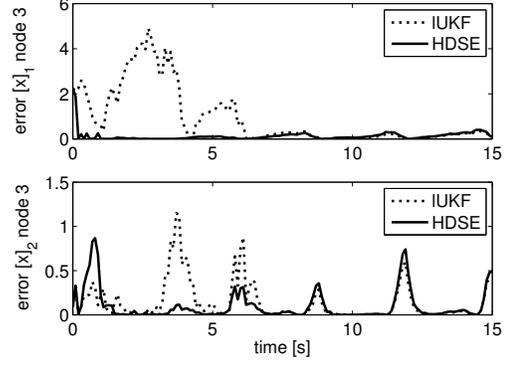


Fig. 6. Squared estimation error in node 3 for both the IUKF and HDSE.

VII. BENCHMARK APPLICATION: TRAFFIC SHOCKWAVES

The traffic shockwave is a spatio-temporal dynamical phenomenon typically emerging from high density highway traffic. It is characterized by an increase in vehicle density and a decrease in vehicle speed. Shockwaves “travel” along the highway upstream (i.e. opposite direction to the traffic). This benchmark consists of initiating a shockwave, after which the goal is to track this (simulated) shockwave using aggregated measurements of speed and density within certain road segments. To that extent, consider a stretch of a one-lane road that is divided into 20 segments of each $L = 500$ meter. A total of 5 nodes are used to monitor shockwaves on that particular road. Every node takes measurements of the average speed and density within its own segment. Node 1 is located at road segment 1, node 2 at segment 5, node 3 at segment 10, node 4 at segment 15 and node 5 at road segment 20. The communication topology of the nodes is similar to Figure 3.

A shockwave, including measurements, are simulated using the discrete-time METANET-model of [14]. Therein, $s^n(t) \in \mathbb{R}$ and $\rho^n(t) \in \mathbb{R}$ denote the average speed and density within road segment n at time t . The METANET-model defines a relation of this average speed and density between neighboring segments, for some $\tau, \eta, \kappa, \rho_{crit}, \alpha, v_{free} \in \mathbb{R}$ and sampling-time δ , i.e.,

$$\begin{aligned} \rho^n(t + \delta) &= \rho^n(t) + \frac{\delta}{L} (\rho^{n-1}(t) s^{n-1}(t) - \rho^n(t) s^n(t)), \\ s^n(t + \delta) &= s^n(t) + \frac{\delta}{\tau} \left(v_{free} e^{-\frac{1}{\alpha} \left(\frac{\rho^n(t)}{\rho_{crit}} \right)^\alpha} - s^n(t) \right) \\ &\quad + \frac{\delta}{L} s^n(t) (s^{n-1}(t) - s^n(t)) - \frac{\eta \delta}{\tau L} \frac{\rho^{n+1}(t) - \rho^n(t)}{\rho^n(t) + \kappa}. \end{aligned}$$

Three configurations of DSEs are employed. All aim at recovering the average speed and density at each segment, based on corresponding measurements at each predefined segment. The first two configurations are the previously described DUKF and the DEKF, in which DEKF employs a EKF as LSE at each node. The third configuration implements the HDSE, which is defined with the following LSEs: nodes 1, 3 and 5 employ an UKF, while nodes 2 and 4 perform an EKF-algorithm. All nodes of all three DSEs

start with equivalent initial estimates, i.e., $s^{(n)}(0) = 85$ and $\rho^n(0) = 30$, for all $n \in \mathbb{Z}_{[1,20]}$.

Notice, that the METANET-model requires values for $\rho^0(t)$, $\rho^{21}(t)$ and $s^0(t)$. Since this information is not available to the DSEs, their values are modeled as process noise. Figure 7 shows the real and estimated density at node 3 according to the DEKF, DUKF and HDSE. The following values were used in this simulation, $\tau = 0.0039$, $\eta = 191$, $\kappa = 254$, $\rho_{crit} = 33.0$, $\alpha = 5.61$, $v_{free} = 89.9$ and $\delta = \frac{10}{3600}$. Estimated values of density at the other nodes are similar to node 3. The results of average speed were omitted to limit the number of pages.

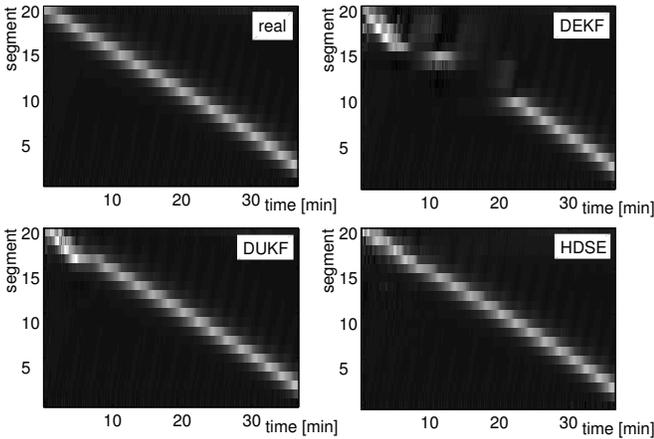


Fig. 7. The real density of all 20 segments in time and their estimated values at node 3 according to the DEKF, DUKF and HDSE. The black color equals a value of 20, while white represents a density of 80 cars per km.

Figure 7 shows that the DEKF suffers from its linearization in the sense that its estimated wave tends to “die out” after it was measured. See for example the wave that is briefly measured at segment 15 around 11 minutes, after which it fades away. Results of the HDSE show that this can be solved by replacing some EKFs in the network with an UKF. Moreover, the HDSE has similar results as the DUKF and they only differ during their initialization, i.e., the first few minutes. However, in the long run the DUKF has a smaller estimation error than the HDSE. More details of the squared estimation error of these three DSE configurations is found in Figure 8.

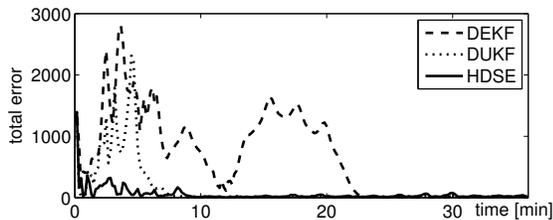


Fig. 8. Summation of the squared estimation-error of the density at node 3, at all the segments and at each sample instant, i.e., $\sum_{n=1}^{20} (\hat{\rho}_3^n(t) - \rho^n(t))^2$.

VIII. CONCLUSIONS

This paper proposed and analyzed a sensor network where different nodes in the network perform different types of local state-estimators. The scheme allows unrestricted combinations of classical Kalman filtering, extended Kalman filtering and unscented Kalman filtering algorithms to calculate the local estimate. The resulting local estimate is then fused with the estimates from neighboring nodes in a local state-fusion method. The proposed distributed estimation set-up could incorporate new nodes (or loose existing ones) on the fly, resulting in a flexible and robust state estimation solution. The benefits of such heterogeneous set-ups were shown in an illustrative example of the Van-der-Pol oscillator and in a benchmark application on shockwaves. These examples showed a decrease in estimation error of nodes that employ the (extended) Kalman filter in case some nodes perform the unscented Kalman filtering algorithm instead. This provides a solid base to further investigate the capabilities of heterogeneous, distributed state-estimation.

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] R. Kandepu, B. Foss, and L. Imsland, “Applying the unscented Kalman filter for nonlinear state estimation,” *Journal of Process Control*, doi:10.1016/j.procont.2007.11.004, 2008.
- [3] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” in *The Proceedings of AeroSense: The 11th International Symposium on Aerospace.*, Orlando, FL, USA, 1997, pp. 182–193.
- [4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: a survey,” *Elsevier, Computer Networks*, vol. 38, pp. 393–422, 2002.
- [5] 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.
- [6] 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.
- [7] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *46th IEEE Conf. on Decision and Control*, New Orleans, LA, USA, 2007, pp. 5492 – 5498.
- [8] J. Sijs, M. Lazar, P. Van de Bosch, and Z. Papp, “An overview of non-centralized Kalman filters,” in *Proceedings of the IEEE Int. Conference on Conference on Control Applications*, San Antonio, USA, 2008, pp. 739–744.
- [9] 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.
- [10] J. Sijs, M. Lazar, and P. v.d. Bosch, “State fusion with unknown correlation: Ellipsoidal intersection,” in *Proceedings of the 2010 American Control Conference*, 2010, pp. 3992 – 3997.
- [11] S. J. Julier and J. K. Uhlmann, “A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations,” in *Proceedings of the American Control Conference*, Piscataway, NJ, USA, 1997, pp. 2369–2373.
- [12] D. Franken and A. Hupper, “Improved Fast Covariance Intersection for Distributed Data Fusion,” in *Proceedings of the 8th Int. Conf. on Information Fusion*, d.o.i.: 10.1109/ICIF.2005.1591849, Philadelphia, PA, USA, 2005.
- [13] L. Chen, P. Arambel, and R. Mehra, “Fusion under Unknown Correlation - Covariance Intersection as a Special Case,” in *Proceedings of 5th IEEE Int. Conf. on Information Fusion*, 2002, pp. 905–912.
- [14] A. Hegyi, B. De Schutter, and H. Hellendoorn, “Model predictive control for optimal coordination of ramp metering and variable speed limits,” *Transportation Research Part C*, vol. 13, no. 3, pp. 185–209, June 2005.