

Distributed Kalman Filtering under Model Uncertainty

Mattia Zorzi

Abstract

We study the problem of distributed Kalman filtering for sensor networks in the presence of model uncertainty. More precisely, we assume that the actual state-space model belongs to a ball, in the Kullback-Leibler topology, about the nominal state-space model and whose radius reflects the mismatch modeling budget allowed for each time step. We propose a distributed Kalman filter with diffusion step which is robust with respect to the aforementioned model uncertainty. Moreover, we derive the corresponding least favorable performance. Finally, we check the effectiveness of the proposed algorithm in the presence of uncertainty through a numerical example.

Index Terms

Distributed robust Kalman filtering, sensor networks (SNs), least favorable analysis.

I. INTRODUCTION

Significant advances in science and technology have led to a large number of problems that involve numerous sensors, i.e. a sensor network (SN), taking measurements and a state process which needs to be estimated from such measurements. Just to mention a few of these problems: area surveillance, region monitoring, target tracking and electrical power grid analysis. These problems can be (in principle) solved by using Kalman filtering equipped with all the observations coming from the SN. On the other hand, such centralized strategy is impractical or impossible to implement. Indeed, it requires a large amount of energy for communications among the central node, i.e. the one which computes an estimate of the state process, and the sensors. In order to overcome this difficulty, distributed strategies have gained rapidly increasing interest in the last few years, see for instance [1], [2], [3], [4]. The latter represent an attractive alternative because they require fewer communications and allow parallel processing. The simpleminded distributed version of the Kalman filter assumes that each node can compute an estimate of the state by using only the observations coming from its neighbors. On the other hand, such an approach provides poor performances compared to the ones of the centralized approach. A remarkable improvement has been gained by the so called consensus-based distributed Kalman filters [5], [6], [7], [8]. Such approaches require multiple communication iterations during each sampling time interval: for instance, in the first iteration the nodes exchange their observation and compute their local estimate; in the second iteration the nodes exchange their local estimates

M. Zorzi is with the Department of Information Engineering, University of Padova, Padova, Italy; e-mail: zorzi@dei.unipd.it

and construct the final estimate based on consensus schemes. A further improvement has been given by diffusion-based strategies, [9], [10], wherein the consensus law is replaced by a convex combination of the local estimates. It is worth noting that there also exist high performing distributed Kalman filters based on different principles. For instance it is possible to compute the estimate through a fusion center which merges the local estimates of the nodes, [11], [12], [13], [14].

Kalman filtering is based on nominal state-space models. On the other hand, the latter are just an approximation of the underlying system, thus the resulting estimate could lead to poor performances in practice. To address this model uncertainty issue many robust Kalman filtering strategies have been proposed. The most popular one is risk-sensitive filtering, [15], [16], wherein large errors are severely penalized according to the so called risk sensitivity parameter: the larger the latter is the more large errors are penalized. Here, we consider the robust Kalman filter introduced by Levy and Nikoukhah [17], [18]. In such approach, the actual state space model belongs to a ball, according to the Kullback-Leibler divergence, about the nominal model and with radius, say tolerance, which represents the mismatch modeling budget allowed for each time step. Then, the optimal estimator is designed according to the least favorable model in this ball. It turns out that the latter obey to a Kalman-like recursion. Furthermore, it can be understood as a generalization of risk-sensitive filtering: such a filter can be rewritten as a risk-sensitive filter with a time varying risk-sensitivity parameter. Finally, it worth noting that the aforementioned approach can be extended to a family of balls formed by using the τ -divergence, [19], [20].

It is then natural to wonder how to perform distributed Kalman filtering for SNs in the presence of model uncertainty. Many papers consider model uncertainty in terms of missing observations. Such a situation well describes communication problems among sensors: for instance, an H_∞ -consensus problem for SNs with multiple missing measurements has been considered in [21]. On the other hand, only few papers addressed the problem of model uncertainty in a broader sense to the best of the author's knowledge. For instance, a distributed Kalman filtering fusion strategy with random state transition and measurements matrices has been considered in [22].

The contribution of the present is about distributed Kalman filtering for SNs wherein the model uncertainty is characterized by a ball, in the Kullback-Leibler divergence topology, about the nominal model. More precisely, we propose a distributed version of the robust Kalman filter introduced by Levy and Nikoukhah. The proposed algorithm is a robust version of the distributed Kalman filter with diffusion step in [9]. Then, we derive the least favorable performance of these filters. Similarly to the centralized case, the latter can be characterized over a finite simulation horizon as follows: first, a forward recursion is required to compute the optimal gains of centralized Kalman filter; then, a backward recursion is required to compute the least favorable model; finally, a forward recursion is required to compute the performance of the distributed robust Kalman algorithm. We show that the average least favorable mean square deviation across the network converges to a finite constant value provided that: the tolerance is sufficiently small; reachability and local observability hold. Moreover, we show that it is very likely that the proposed robust distributed filters perform better than the standard ones provided that the tolerance is sufficiently large. Finally, we compare the proposed algorithms with the standard ones through a numerical example.

The outline of the paper is as follows. In Section II we introduce the background about robust Kalman filtering. In Section III we present the distributed robust Kalman filtering algorithm. In Section IV we analyze the least

favorable performance of the proposed algorithms. In Section V we present a numerical example which compares the proposed algorithm with the standard ones. Finally, in Section VI we draw the conclusions.

II. BACKGROUND

Consider the nominal state-space model

$$\begin{aligned} x_{t+1} &= Ax_t + \Gamma_B u_t \\ y_t &= Cx_t + \Gamma_D u_t \end{aligned} \quad (1)$$

where $x_t \in \mathbb{R}^n$ is the state process, $y_t \in \mathbb{R}^{pN}$ is the observation process, u_t is normalized white Gaussian noise (WGN), i.e. $\mathbb{E}[u_t u_s^T] = I \delta_{t-s}$ where δ_t denotes the Kronecker delta function. We assume that u_t is independent of the initial state x_0 . The latter is Gaussian distributed with mean \hat{x}_0 and covariance matrix V_0 . Model (1) is characterized by the nominal transition probability density of $z_t := [x_{t+1}^T y_t^T]^T$ given x_t which is denoted by $\phi_t(z_t|x_t)$. We assume that $\Gamma_B \Gamma_D^T = 0$, i.e. the noise entering in the state process is independent of the noise entering in the observation process. We assume that u_t affects all the components of the dynamics and observations in (1). Such assumption is necessary whenever entropy-like indexes are used to measure the proximity of statistical models, as in our case, otherwise these indexes take infinite value. Accordingly, the matrix $[\Gamma_B^T \Gamma_D^T]^T$ is full row rank, and without loss of generality we can assume that $[\Gamma_B^T \Gamma_D^T]^T$ is a square (and thus invertible) matrix of dimension $pN + n$. Indeed, we can always compress the column space of such a matrix and remove the noise components which do not affect model in (1). Accordingly, the state-space model in (1) is reachable; moreover we also assume it is observable.

Let $\tilde{\phi}_t(z_t|x_t)$ be the (unknown) actual transition probability density of z_t given x_t . In order to account the fact that the nominal model does not coincide with the actual model, we assume that $\tilde{\phi}_t$ belongs to the closed ball about ϕ_t :

$$\mathbb{B}_t := \{ \tilde{\phi}_t \text{ s.t. } \tilde{\mathbb{E}}[\log(\tilde{\phi}_t/\phi_t)|Y_{t-1}] \leq c \} \quad (2)$$

where

$$\begin{aligned} &\tilde{\mathbb{E}}[\log(\tilde{\phi}_t/\phi_t)|Y_{t-1}] \\ &:= \int \int \tilde{\phi}_t(z_t|x_t) \tilde{f}_t(x_t|Y_{t-1}) \log \left(\frac{\tilde{\phi}_t(z_t|x_t)}{\phi_t(z_t|x_t)} \right) dz_t dx_t \end{aligned} \quad (3)$$

and $Y_{t-1} := \{y_s, s = 1 \dots t-1\}$. The latter represents the relative entropy between the actual and the nominal transition densities $\tilde{\phi}_t(z_t|x_t)$ and $\phi_t(z_t|x_t)$ at time t , respectively, and $\tilde{f}_t(x_t|Y_{t-1}) \sim \mathcal{N}(\hat{x}_t, V_t)$ is the actual conditional probability density of x_t given the past observations Y_{t-1} . Finally, parameter $c > 0$ is called tolerance and represents the mismatch modeling budget allowed for each time step.

Given the nominal model in (1), a robust estimator of x_{t+1} given Y_t is obtained by solving the following mini-max problem

$$\hat{x}_{t+1} = \operatorname{argmin}_{g_t \in \mathbb{G}_t} \max_{\tilde{\phi}_t \in \mathbb{B}_t} \tilde{\mathbb{E}}[\|x_{t+1} - g_t\|^2 | Y_{t-1}] \quad (4)$$

where

$$\begin{aligned} & \tilde{\mathbb{E}}[\|x_{t+1} - g_t\|^2 | Y_{t-1}] \\ & := \int \int \|x_{t+1} - g_t\|^2 \tilde{\phi}_t(z_t | x_t) \tilde{f}_t(x_t | Y_{t-1}) dx_t dz_t \end{aligned} \quad (5)$$

represents the mean square error of the estimator g_t which is a function of y_t and Y_{t-1} . \mathbb{G}_t denotes the set of all estimators g_t such that $\tilde{\mathbb{E}}[\|g_t\|^2]$ is finite for any $\tilde{\phi}_t \in \mathbb{B}_t$. Roughly speaking, such estimator is designed according to the least favorable model whose mismodeling budget allowed is expressed at each time step. This way to characterize model uncertainty is better than expressing the uncertainty over the entire simulation interval. Indeed, in the latter case the maximizer has the possibility to identify the moment where the dynamic of model (1) is most susceptible to distortion and to allocate most of the distortion budget specified by the tolerance to this single element of the model, that is a situation which is pretty unrealistic.

In [18] it has been proved that the estimator solving the mini-max problem (4) obeys the Kalman-like recursion:

$$\begin{aligned} G_t &= AV_t C^T (CV_t C^T + \Gamma_D \Gamma_D^T)^{-1} \\ \hat{x}_{t+1} &= A\hat{x}_t + G_t(y_t - C\hat{x}_t) \\ P_{t+1} &= A(V_t^{-1} + C^T(\Gamma_D \Gamma_D^T)^{-1}C)^{-1}A^T + \Gamma_B \Gamma_B^T \\ \text{Find } \theta_t \text{ s.t. } \gamma(P_{t+1}, \theta_t) &= c \\ V_{t+1} &= (P_{t+1}^{-1} - \theta_t I)^{-1} \end{aligned} \quad (6)$$

where

$$\gamma(P, \theta) := \log \det(I - \theta P) + \text{tr}((I - \theta P)^{-1} - I). \quad (7)$$

The so called risk sensitivity parameter, [23], $\theta_t > 0$ does always exist and it is unique given P_{t+1} and c , moreover it can be computed efficiently by using a bisection algorithm. In the limit case $c = 0$, i.e. there is no uncertainty, then $\theta_t = 0$ and (6) becomes the usual Kalman filter.

Remark 1: The robust filtering paradigm in (4) can be extended to nominal state-space models with time-varying parameters and tolerance. On the other hand, to ease the introduction of the corresponding distributed algorithms we stick to the constant parameters and tolerance case.

III. DISTRIBUTED ROBUST KALMAN FILTER

Consider a network of N nodes and in each node there is one sensor. We say that two nodes are connected if the corresponding sensors can communicate directly with each other. A node is always connected with itself. The neighborhood of node k , i.e. the set of nodes connected with k , is denoted by \mathcal{N}_k , in particular $k \in \mathcal{N}_k$. The corresponding $N \times N$ adjacency matrix $J = [j_{lk}]_{lk}$ is defined as

$$j_{lk} := \begin{cases} 1, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

The number of neighbors of node k is denoted by n_k . Every node at time t collects a measurement $y_{k,t} \in \mathbb{R}^p$ whose underlying model is unknown. The nominal model takes the form:

$$\begin{aligned} x_{t+1} &= Ax_t + Bw_t \\ y_{k,t} &= C_k x_t + D_k v_{k,t} \quad k = 1 \dots N \end{aligned} \quad (9)$$

where w_t and $v_{k,t}$ $k = 1 \dots N$ are independent WGNs such that $\mathbb{E}[w_t w_t^T] = I$, $\mathbb{E}[w_t v_{k,t}^T] = 0$ and $\mathbb{E}[v_{l,t} v_{k,s}^T] = I \delta_{k-l} \delta_{t-s}$. It is worth noting that (9) can be rewritten as (1) with $y_t = [y_{1,t}^T \dots y_{N,t}^T]^T$, $u_t = [w_t^T \ v_t^T]^T$, $v_t = [v_{1,t}^T \dots v_{N,t}^T]^T$, $\Gamma_B = [B \ 0]$, $\Gamma_D = [0 \ D]$,

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_N \end{bmatrix}, \quad D = \text{diag}(D_1, \dots, D_N) \quad (10)$$

where diag is the linear operator which constructs a block diagonal matrix whose blocks are the ones specified in the argument. We also define $R := DD^T$, $R_l := D_l D_l^T$ with $l = 1 \dots N$, and

$$S_{tot} := C^T R^{-1} C = \sum_{l=1}^N C_l^T R_l^{-1} C_l. \quad (11)$$

Accordingly, the filtering gain for model (9) can be written as

$$\begin{aligned} G_t &= A V_t C^T (C V_t C^T + R)^{-1} = A V_t C^T \times \\ &\quad \times (R^{-1} - R^{-1} C (V_t^{-1} + C^T R^{-1} C)^{-1} C^T R^{-1}) \\ &= A (V_t^{-1} + C^T R^{-1} C)^{-1} C^T R^{-1} \\ &= A (V_t^{-1} + S_{tot})^{-1} C^T R^{-1} \end{aligned}$$

and thus

$$\begin{aligned} G_t y_t &= A (V_t^{-1} + S_{tot})^{-1} \sum_{l=1}^N C_l^T R_l^{-1} y_{l,t} \\ G_t C &= (V_t^{-1} + S_{tot})^{-1} S_{tot}. \end{aligned}$$

In distributed Kalman filtering under model uncertainty, the aim is to compute for every node k a prediction of the state x_t while sharing the data only with its neighbors $l \in \mathcal{N}_k$ and taking into account that (9) does not coincide with the actual model. In what follows, the one step-ahead prediction of x_t at node k is denoted by $\hat{x}_{k,t}$. It is not difficult to see that the robust Kalman filter for model (9), i.e. the node k has access to all measurements across

all the nodes in the network, can be written as

$$\begin{aligned}
\hat{x}_{k,t+1} &= A\hat{x}_{k,t} + A(V_{k,t}^{-1} + S_{tot})^{-1} \times \\
&\quad \times \sum_{l=1}^N C_l^T R_l^{-1} (y_{l,t} - C_l \hat{x}_{k,t}) \\
P_{k,t+1} &= A \left(V_{k,t}^{-1} + S_{tot} \right)^{-1} A^T + B B^T \\
\text{Find } \theta_{k,t} \text{ s.t. } &\gamma(P_{k,t+1}, \theta_{k,t}) = c \\
V_{k,t+1} &= (P_{k,t+1}^{-1} - \theta_{k,t} I)^{-1}
\end{aligned} \tag{12}$$

where $\hat{x}_{k,t} = \hat{x}_t$ and $V_{k,t} = V_t$ with $k = 1 \dots N$.

Therefore in the case that the node k has not access to all measurements across all nodes in the network, we would obtain a state prediction $\hat{x}_{k,t}$ of x_t which is as close as to the global state prediction.

A. Robust Kalman filter with diffusion step

We assume that a node k has access to the measurements of its neighbors \mathcal{N}_k . The corresponding nominal state-space model is

$$\begin{aligned}
x_{t+1} &= A x_t + B w_t \\
y_{l,t} &= C_l x_t + D_l v_{l,t}, \quad l \in \mathcal{N}_k.
\end{aligned} \tag{13}$$

The latter can be rewritten as a state-space model $(A, \Gamma_B, C_k^{loc}, \Gamma_{D_k^{loc}})$ with input noise $u_{k,t}^{loc} = [w_t^T \ (v_{k,t}^{loc})^T]^T$ and output $y_{k,t}^{loc}$ where $v_{k,t}^{loc}$ and $y_{k,t}^{loc}$ are obtained by stacking $v_{l,t}$ and $y_{l,t}$, respectively, with $l \in \mathcal{N}_k$. Moreover, $\Gamma_B = [B \ 0]$, C_k^{loc} is obtained by stacking C_l with $l \in \mathcal{N}_k$, $\Gamma_{D_k^{loc}} = [0 \ D_k^{loc}]$ and D_k^{loc} is a block diagonal matrix whose main blocks are D_l with $l \in \mathcal{N}_k$. We also define $R_k^{loc} := D_k^{loc} (D_k^{loc})^T$, $S_k := (C_k^{loc})^T (R_k^{loc})^{-1} C_k^{loc}$ and thus

$$S_k = \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} C_l. \tag{14}$$

Accordingly, the one-step ahead predictor $\hat{x}_{k,t}$ of x_t at node k is given by (12) where the terms for which $l \notin \mathcal{N}_k$ are discarded. Then, the local prediction $\hat{x}_{k,t+1}$ can be understood as an intermediate local prediction of x_t at node k . In what follows we denote such intermediate prediction as $\psi_{k,t+1}$. Then, the idea is to update the prediction at node k not only in terms of $\psi_{k,t+1}$, but also in terms of $\psi_{l,t+1}$ with $l \in \mathcal{N}_k$. More precisely, we consider a matrix $W = [w_{lk}]_{lk} \in \mathbb{R}^{N \times N}$ such that

$$\begin{aligned}
w_{lk} &\geq 0 \text{ and } w_{lk} = 0 \text{ if } l \notin \mathcal{N}_k \\
\sum_{l \in \mathcal{N}_k} w_{lk} &= 1.
\end{aligned} \tag{15}$$

Then, the final prediction at node k is obtained by the so called diffusion step, [9]:

$$\hat{x}_{k,t+1} = \sum_{l \in \mathcal{N}_k} w_{lk} \psi_{l,t+1}. \tag{16}$$

Therefore, in the diffusion algorithm a node k exploits the information of the neighbors in terms of $y_{l,t}$ and $\psi_{l,t+1}$. The resulting procedure is outlined in Algorithm 1. As explained in [9], the diffusion step (16) is motivated by

Algorithm 1: Distributed Robust Kalman filter with diffusion step at time t

Input : $\hat{x}_{k,t}, V_{k,t}, y_{k,t}$ with $k = 1 \dots N$

Output: $\hat{x}_{k,t+1}, V_{k,t+1}$ with $k = 1 \dots N$

Incremental step. Compute at every node k :

$$\begin{aligned} \psi_{k,t+1} &= \\ &A\hat{x}_{k,t} + A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} (y_{l,t} - C_l \hat{x}_{k,t}) \\ P_{k,t+1} &= A(V_{k,t}^{-1} + S_k)^{-1} A^T + BB^T \\ \text{Find } \theta_{k,t} \text{ s.t. } &\gamma(P_{k,t+1}, \theta_{k,t}) = c \\ V_{k,t+1} &= (P_{k,t+1}^{-1} - \theta_{k,t} I)^{-1} \end{aligned}$$

Diffusion step. Compute at every node k :

$$\hat{x}_{k,t+1} = \sum_{l \in \mathcal{N}_k} w_{lk} \psi_{l,t+1}$$

the fact that the centralized prediction \hat{x}_{t+1} can be approximated by a local convex combination of $\psi_{l,t+1}$. It is worth noting that in the case that $c = 0$, i.e. there is no mismatch between the actual and the nominal model, then $\theta_{k,t} = 0$ for any t and k so that, we obtain the diffusion algorithm proposed in [9]. In the case that

$$w_{lk} := \begin{cases} \varepsilon, & \text{if } l \neq k, l \in \mathcal{N}_k \\ 1 - \varepsilon(n_k - 1), & \text{if } l = k, l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

we obtain a consensus-based update where $\varepsilon > 0$ is the consensus parameter. Indeed, in the case that $c = 0$ and W is designed as in (17) we obtain the distributed consensus-based algorithm proposed in [7]. Finally, in the case that $W = I$ we obtain the robust version of the local Kalman filter [24, p. 329].

It is worth noting that the mismatch modeling budget c in Algorithm 1 coincides with the one of the centralized filter. Such a choice does not guarantee that the least favorable model computed at node k coincides with the one of the centralized filter. On the other hand, we will see that, under large deviations of the least favorable model of the centralized problem, it is very likely that the predictor at node k using Algorithm 1 performs better than the one in [9], see Section IV-B for more details.

Remark 2: In some cases we may have a state-space model of the form

$$\begin{aligned} x_{t+1} &= Ax_t + \Gamma_B u_t + r_t \\ y_t &= Cx_t + \Gamma_D u_t \end{aligned}$$

where r_t is a deterministic process. In [25] it was shown that the corresponding centralized robust Kalman filter still obeys the Kalman-like recursion (6) where the prediction update is replaced by

$$\hat{x}_{t+1} = A\hat{x}_t + G_t(y_t - C\hat{x}_t) + r_t. \quad (18)$$

Then, it is not difficult to see that the distributed algorithms presented in this section still hold in this case. The unique difference is that we need to add in the prediction update of each node the term r_t .

IV. LEAST FAVORABLE PERFORMANCE

In this section we analyze the performance of the distributed algorithm with diffusion step introduced in Section III under the least favorable model which is solution of the mini-max problem (4), i.e. the centralized problem. The performance assessment is given by the mean and variance of the least favorable state prediction error for each node k (including the diffusion step), say $\tilde{x}_{k,t}$ with $k = 1 \dots N$. In [18], [20] it has been shown that the least favorable model can be characterized over a finite interval $[0, T]$ and it takes the following form:

$$\begin{aligned}\xi_{t+1} &= \check{A}_t \xi_t + \check{B}_t \varepsilon_t \\ y_t &= \check{C}_t \xi_t + \check{D}_t \varepsilon_t\end{aligned}\tag{19}$$

where $\xi_t = [x_t^T \ e_t^T]^T$, x_t is the least favorable state process, e_t is the least favorable prediction error of x_t using the robust filter (6) and ε_t is WGN with covariance matrix equal to the identity. Moreover,

$$\begin{aligned}\check{A}_t &:= \begin{bmatrix} A & \Gamma_B \Gamma_{H_t} \\ 0 & A - G_t C + (\Gamma_B - G_t \Gamma_D) \Gamma_{H_t} \end{bmatrix} \\ \check{B}_t &:= \begin{bmatrix} \Gamma_B \Gamma_{L_t} \\ (\Gamma_B - G_t \Gamma_D) \Gamma_{L_t} \end{bmatrix} \\ \check{C}_t &:= \begin{bmatrix} C & \Gamma_D \Gamma_{H_t} \end{bmatrix}, \quad \check{D}_t := \Gamma_D \Gamma_{L_t},\end{aligned}\tag{20}$$

where Γ_{L_t} is such that $K_t = \Gamma_{L_t} \Gamma_{L_t}^T$,

$$\begin{aligned}K_t &:= (I - (\Gamma_B - G_t \Gamma_D)^T (\Omega_{t+1}^{-1} + \theta_t I) (\Gamma_B - G_t \Gamma_D))^{-1} \\ \Gamma_{H_t} &:= K_t (\Gamma_B - G_t \Gamma_D)^T (\Omega_{t+1}^{-1} + \theta_t I) (A - G_t C).\end{aligned}$$

Matrix Ω_{t+1}^{-1} is computed from the backward recursion

$$\begin{aligned}\Omega_t^{-1} &= (A - G_t C)^T (\Omega_{t+1}^{-1} + \theta_t I) (A - G_t C) \\ &\quad + \Gamma_{H_t}^T K_t^{-1} \Gamma_{H_t}\end{aligned}\tag{21}$$

where the final point is initialized with $\Omega_{T+1}^{-1} = 0$ and T is the simulation horizon. Therefore, to construct the least favorable model we need to compute the gain G_t performing a forward sweep of the robust Kalman filter in (6) over the interval $[0, T]$, then we generate the matrices Ω_t through a backward sweep over the interval $[0, T]$. We partition $\Gamma_{H_t} \in \mathbb{R}^{(pN+n) \times n}$ and $\Gamma_{L_t} \in \mathbb{R}^{(pN+n) \times (pN+n)}$ as $\Gamma_{H_t} = [M_t^T \ H_t^T]^T$ and $\Gamma_{L_t} = [N_t^T \ L_t^T]^T$ with $M_t \in \mathbb{R}^{n \times n}$, $H_t \in \mathbb{R}^{pN \times n}$, $N_t \in \mathbb{R}^{n \times (pN+n)}$ and $L_t \in \mathbb{R}^{pN \times (pN+n)}$. Moreover, we partition H_t and L_t as follows:

$$H_t = \begin{bmatrix} H_{1,t} \\ \vdots \\ H_{N,t} \end{bmatrix}, \quad L_t = \begin{bmatrix} L_{1,t} \\ \vdots \\ L_{N,t} \end{bmatrix}\tag{22}$$

where $H_{k,t} \in \mathbb{R}^{p \times n}$ and $L_{k,t} \in \mathbb{R}^{p \times (pN+n)}$.

Next we express the least favorable state prediction error $\tilde{x}_{k,t}$ at node k in terms of the WGN ε_t : in this way we will be able to characterize the mean and the variance of $\tilde{x}_{k,t}$. We define

$$\begin{aligned}\tilde{x}_{k,t} &= x_t - \hat{x}_{t,k} \\ \tilde{\psi}_{k,t} &= x_t - \psi_{k,t}\end{aligned}\tag{23}$$

which represent the prediction error and the intermediate prediction error, respectively, at node k at time t . Notice that

$$y_{l,t} = [C_l \ D_l H_{l,t}] \xi_t + D_l L_{l,t} \varepsilon_t\tag{24}$$

for $l = 1 \dots N$.

Therefore, we have

$$\begin{aligned}
\tilde{\psi}_{k,t+1} &= x_{t+1} - \psi_{k,t+1} = [I \ 0]\xi_{t+1} - \psi_{k,t+1} \\
&= [I \ 0](\check{A}_t\xi_t + \check{B}_t\varepsilon_t) - A\hat{x}_{k,t} \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} (y_{l,t} - C_l \hat{x}_{k,t}) \\
&= Ax_t + \Gamma_B \Gamma_{H_t} e_t + \Gamma_B \Gamma_{L_t} \varepsilon_t - A\hat{x}_{k,t} \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} ([C_l \ D_l H_{l,t}] \xi_t \\
&\quad + D_l L_{l,t} \varepsilon_t - C_l \hat{x}_{k,t}) \\
&= A\tilde{x}_{k,t} + BM_t e_t + BN_t \varepsilon_t \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} (C_l x_t + D_l H_{l,t} e_t \\
&\quad + D_l L_{l,t} \varepsilon_t - C_l \hat{x}_{k,t}) \\
&= A\tilde{x}_{k,t} + BM_t e_t + BN_t \varepsilon_t \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} (C_l \tilde{x}_{k,t} + D_l H_{l,t} e_t \\
&\quad + D_l L_{l,t} \varepsilon_t) \\
&= A(I - (V_{k,t}^{-1} + S_k)^{-1} S_k) \tilde{x}_{k,t} + B(M_t e_t + N_t \varepsilon_t) \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} D_l (H_{l,t} e_t + L_{l,t} \varepsilon_t) \\
&= A(V_{k,t}^{-1} + S_k)^{-1} V_{k,t}^{-1} \tilde{x}_{k,t} + B(M_t e_t + N_t \varepsilon_t) \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} D_l (H_{l,t} e_t + L_{l,t} \varepsilon_t) \\
&= A(V_{k,t}^{-1} + S_k)^{-1} V_{k,t}^{-1} \tilde{x}_{k,t} + B(M_t e_t + N_t \varepsilon_t) \\
&\quad - A(V_{k,t}^{-1} + S_k)^{-1} \sum_{l=1}^N j_{lk} C_l^T R_l^{-1} D_l (H_{l,t} e_t + L_{l,t} \varepsilon_t) \tag{25}
\end{aligned}$$

where we recall that $j_{lk} = 1$ if $l \in \mathcal{N}_k$, otherwise $j_{lk} = 0$. Notice that,

$$\begin{aligned}
\tilde{x}_{k,t+1} &= x_t - \sum_{l \in \mathcal{N}_k} w_{lk} \psi_{l,t+1} \\
&= \sum_{l=1}^N w_{lk} x_t - \sum_{l=1}^N w_{lk} \psi_{l,t+1} \\
&= \sum_{l=1}^N w_{lk} (x_t - \psi_{l,t+1}) = \sum_{l=1}^N w_{lk} \tilde{\psi}_{l,t+1} \tag{26}
\end{aligned}$$

where we have exploited (15). Taking into account (25), we have

$$\begin{aligned}\tilde{x}_{k,t+1} &= \sum_{l=1}^N w_{lk} [A(V_{l,t}^{-1} + S_l)^{-1} V_{l,t}^{-1} \tilde{x}_{l,t} - A(V_{l,t}^{-1} + S_l)^{-1} \\ &\quad \times \sum_{m=1}^N j_{ml} C_m^T R_m^{-1} D_m (H_{m,t} e_t + L_{m,t} \varepsilon_t)] \\ &\quad + B(M_t e_t + N_t \varepsilon_t).\end{aligned}\tag{27}$$

By defining

$$\begin{aligned}\tilde{\chi}_t &:= [\tilde{x}_{1,t}^T \dots \tilde{x}_{N,t}^T]^T \\ \mathcal{C} &:= \text{diag}(C_1, \dots, C_N) \\ \mathcal{V}_t &:= \text{diag}(V_{1,t}, \dots, V_{N,t}) \\ \mathcal{S} &:= \text{diag}(S_1, \dots, S_N)\end{aligned}\tag{28}$$

we can rewrite (27) in the following compact way:

$$\begin{aligned}\tilde{x}_{k,t+1} &= ([w_{1k} \dots w_{Nk}] \otimes I) \{ (I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} \\ &\quad \times \mathcal{V}_t^{-1} \tilde{\chi}_t - (I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} (J^T \otimes I) \\ &\quad \times \mathcal{C}^T R^{-1} D(H_t e_t + L_t \varepsilon_t) \} + B(M_t e_t + N_t \varepsilon_t)\end{aligned}$$

and thus

$$\begin{aligned}\tilde{\chi}_{t+1} &= (W^T \otimes I) \{ (I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} \mathcal{V}_t^{-1} \tilde{\chi}_t \\ &\quad - (I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} (J^T \otimes I) \mathcal{C}^T R^{-1} D \\ &\quad \times (H_t e_t + L_t \varepsilon_t) \} + (\mathbf{1} \otimes B)(M_t e_t + N_t \varepsilon_t)\end{aligned}\tag{29}$$

where $\mathbf{1}$ denotes the vector of ones. We rewrite the latter as

$$\tilde{\chi}_{t+1} = \mathcal{A}_t \tilde{\chi}_t + \mathcal{B}_t \varepsilon_t + \mathcal{C}_t e_t\tag{30}$$

where

$$\begin{aligned}\mathcal{A}_t &:= (W^T \otimes I)(I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} \mathcal{V}_t^{-1} \\ \mathcal{B}_t &:= -(W^T \otimes I)(I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} (J^T \otimes I) \mathcal{C} R^{-1} D L_t \\ &\quad + \mathbf{1} \otimes B N_t \\ \mathcal{C}_t &:= -(W^T \otimes I)(I \otimes A)(\mathcal{V}_t^{-1} + \mathcal{S})^{-1} (J^T \otimes I) \mathcal{C} R^{-1} D H_t \\ &\quad + \mathbf{1} \otimes B M_t.\end{aligned}\tag{31}$$

Combining (30) with the model for e_t in (19), we obtain:

$$\eta_{t+1} = \mathcal{F}_t \eta_t + \mathcal{G}_t \varepsilon_t\tag{32}$$

where $\eta_t := [\tilde{\chi}_t^T \ e_t^T]^T$,

$$\begin{aligned} \mathcal{F}_t &:= \begin{bmatrix} \mathcal{A}_t & \mathcal{C}_t \\ 0 & (A - G_t C) + (\Gamma_B - G_t \Gamma_D) \Gamma_{H_t} \end{bmatrix} \\ \mathcal{G}_t &:= \begin{bmatrix} \mathcal{B}_t \\ (\Gamma_B - G_t \Gamma_D) \Gamma_{L_t} \end{bmatrix}. \end{aligned} \quad (33)$$

From (32), we can analyze the performance of the distributed Algorithm 1. Taking the expectation of (32), we obtain

$$\tilde{\mathbb{E}}[\eta_{t+1}] = \mathcal{F}_t \tilde{\mathbb{E}}[\eta_t]. \quad (34)$$

Since \hat{x}_0 is the mean of x_0 and $\hat{x}_{k,0} = \hat{x}_0$ for $k = 1 \dots N$, we have that $\tilde{\mathbb{E}}[\eta_0] = 0$. Accordingly, η_t is a random vector with zero mean for any t . This means that the distributed Kalman predictions with diffusion step are unbiased.

We proceed to analyze the variance of the prediction errors. We define $\mathcal{Q}_t = \tilde{\mathbb{E}}[\eta_t \eta_t^T]$. Since ε_t is WGN with covariance matrix equal to the identity, by (32) we have that \mathcal{Q}_t is given by solving the following Lyapunov equation

$$\mathcal{Q}_{t+1} = \mathcal{F}_t \mathcal{Q}_t \mathcal{F}_t^T + \mathcal{G}_t \mathcal{G}_t^T. \quad (35)$$

We partition \mathcal{Q}_t as follows:

$$\mathcal{Q}_t = \begin{bmatrix} \mathcal{P}_t & \mathcal{H}_t \\ \mathcal{H}_t^T & \mathcal{R}_t \end{bmatrix} \quad (36)$$

where $\mathcal{P}_t \in \mathbb{R}^{Np \times Np}$, $\mathcal{H}_t \in \mathbb{R}^{Np \times n}$ and $\mathcal{R}_t \in \mathbb{R}^{n \times n}$. The $n \times n$ matrices in the main block diagonal of \mathcal{P}_t represent the covariance matrices of the estimation error at each node. Let $\text{MSD}_{k,t} := \tilde{\mathbb{E}}[\|x_t - \hat{x}_{k,t}\|^2]$ denote the least favorable mean square deviation at node k and at time t . Then, the average least favorable mean square deviation across the network at time t is

$$\overline{\text{MSD}}_t := \frac{1}{N} \sum_{k=1}^N \text{MSD}_{k,t} = \frac{1}{N} \text{tr}(\mathcal{P}_t). \quad (37)$$

The computation of the sequence \mathcal{P}_t depends on the simulation horizon T . In particular, it is required to perform three steps:

- compute the filtering gain G_t performing a forward sweep of the centralized robust Kalman filter in (6) over the interval $[0, T]$
- compute Ω_t performing the backward recursion (21) over the interval $[0, T]$
- compute \mathcal{P}_t performing a forward sweep of the Lyapunov equation in (35) over the interval $[0, T]$.

A. Convergence analysis

In the previous section we showed how to compute \mathcal{Q}_t over the simulation horizon $[0, T]$. Let $0 < \alpha < \beta < 1$. We show that under reachability and local observability, and choosing the tolerance $c > 0$ sufficiently small, then \mathcal{Q}_t converges over the interval $[\alpha T, \beta T]$ as T approaches infinity, and thus the prediction errors at each node have zero mean and finite constant variance in steady state. It is worth noting that local observability is a strong assumption which may pose some limits on the practical applicability of the distributed robust Kalman filter.

In [26] it has been shown that the least favorable prediction error using an estimator of type $\hat{x}_{t+1} = A\hat{x}_t + \tilde{G}_t(y_t - C\hat{x}_t)$, where C coincides with the one in the least favorable model, has zero mean and convergent covariance matrix provided that c is sufficiently small. However, these results cannot be directly applied to our case because the predictor at node k is given by a convex combination of local estimators whose matrix C does not coincide with the one of the least favorable model.

The update of the intermediate local prediction can be rewritten as

$$\psi_{k,t+1} = A\hat{x}_{k,t} + G_{k,t}(y_{k,t}^{loc} - C_k^{loc}\hat{x}_{k,t}) \quad (38)$$

where

$$G_{k,t} = A(V_{k,t}^{-1} + S_k^{-1})^{-1}(C_k^{loc})^T(R_k^{loc})^{-1} \quad (39)$$

is the filtering gain at node k . The first step is to show that $G_{k,t}$ converges as t approaches infinity.

Proposition 4.1: Assume that the pair (A, B) is reachable and that the pair (A, C_k^{loc}) is observable for every k . Then, there exists $c > 0$ sufficiently small such that for any $V_{k,0} > 0$ the sequence $P_{k,t}$ $t \geq 0$ generated by Algorithm 1 at node k converges to a unique solution $\bar{P}_k > 0$. Furthermore, $\theta_{k,t} \rightarrow \bar{\theta}_k$, $V_{k,t} \rightarrow \bar{V}_k > 0$ and the limit \bar{G}_k of the filtering gain $G_{k,t}$ is such that $A - \bar{G}_k C_k^{loc}$ is Schur stable. Moreover, \bar{P}_k is the unique solution of the algebraic Riccati-like equation

$$\bar{P}_k = A(\bar{P}_k^{-1} - \bar{\theta}_k I + (C_k^{loc})^T(R_k^{loc})^{-1}C_k^{loc})^{-1}A^T + BB^T. \quad (40)$$

Proof: The convergence of the local robust Kalman filter follows from the convergence result of the robust Kalman filter in [27, Proposition 3.5], see also [28], under the assumption that the local state space model $(A, B, C_k^{loc}, D_k^{loc})$ is reachable and observable. \square

Regarding the least favorable model in (19), it is possible to prove that it does converge to a state space model with constant parameters, as the simulation horizon T tends to infinity.

Proposition 4.2 (Zorzi, Levy [26]): Assume that the pair (A, B) is reachable and that the pair (A, C) is observable. Then, there exists $c > 0$ sufficiently small such that:

- the forward sequences G_t and θ_t , $t \geq 0$, of the centralized robust Kalman filter (6) converges to \bar{G} and $\bar{\theta}$, respectively, as t tends to infinity;
- when the simulation horizon T tends to infinity, the backward sequence Ω_t generated by (21), with the steady state parameters \bar{G} and $\bar{\theta}$ of the centralized robust Kalman filter, converges to $\bar{\Omega}$. Furthermore,

$$L_t \rightarrow \bar{L}, \quad K_t \rightarrow \bar{K}, \quad \Gamma_{H_t} \rightarrow \Gamma_{\bar{H}} \quad (41)$$

and $(A - \bar{G}C) + (B - \bar{G}\Gamma_D)\Gamma_{\bar{H}}$ is Schur stable.

Finally, we need of the following result.

Proposition 4.3 (Cattivelli, Sayed [9]): Consider the time-varying Lyapunov equation

$$X_{t+1} = A_t X_t A_t + Q_t \quad (42)$$

where A_t and Q_t converges to A and Q , respectively, as $t \rightarrow \infty$, with A Schur stable. Then, X_t converges to the unique solution X of the Lyapunov equation:

$$X = AXA^T + Q. \quad (43)$$

We are ready to prove the main convergence result.

Proposition 4.4: Assume that the pair (A, B) is reachable and that the pair (A, C_k^{loc}) is observable for every k . Then, there exists $c > 0$ sufficiently small such that for any $V_0 > 0$ and $V_{k,0} > 0$ the sequence \mathcal{Q}_t , $t \geq 0$, generated by (35) converges to $\bar{\mathcal{Q}} > 0$, $\mathcal{F}_t \rightarrow \bar{\mathcal{F}}$ and $\mathcal{G}_t \rightarrow \bar{\mathcal{G}}$ over $[\alpha T, \beta T]$ as $T \rightarrow \infty$. Moreover, $\bar{\mathcal{Q}}$ is the unique solution of

$$\bar{\mathcal{Q}} = \bar{\mathcal{F}}\bar{\mathcal{Q}}\bar{\mathcal{F}}^T + \bar{\mathcal{G}}\bar{\mathcal{G}}^T \quad (44)$$

and $\bar{\mathcal{F}}$ is Schur stable. Therefore, the average least favorable mean square deviation across the network $\overline{\text{MSD}}_t$ does converge over $[\alpha T, \beta T]$ as $T \rightarrow \infty$.

Proof: Notice that the assumptions of Proposition 4.1 hold. Therefore, we have that $\mathcal{V}_t \rightarrow \bar{\mathcal{V}}$ and $\mathcal{A}_t \rightarrow \bar{\mathcal{A}}$ where

$$\begin{aligned} \bar{\mathcal{V}} &:= \text{diag}(\bar{V}_1, \dots, \bar{V}_N) \\ \bar{\mathcal{A}} &:= (W^T \otimes I)(I \otimes A)(\bar{\mathcal{V}}^{-1} + \mathcal{S})^{-1}\bar{\mathcal{V}}^{-1}. \end{aligned} \quad (45)$$

Moreover, $\bar{G}_k = A(\bar{V}_k^{-1} + S_k^{-1})^{-1}(C_k^{loc})^T(R_k^{loc})^{-1}$ and thus

$$\begin{aligned} A - \bar{G}_k C_k^{loc} &= A(I - (\bar{V}_k^{-1} + S_k)^{-1}(C_k^{loc})^T(R_k^{loc})^{-1}C_k^{loc}) \\ &= A(I - (\bar{V}_k^{-1} + S_k)^{-1}S_k) \\ &= A(\bar{V}_k^{-1} + S_k)^{-1}\bar{V}_k^{-1} \end{aligned} \quad (46)$$

which is Schur stable. Accordingly, we have that the block-diagonal matrix

$$\mathcal{M} := (I \otimes A)(\bar{\mathcal{V}}^{-1} + \mathcal{S})^{-1}\bar{\mathcal{V}}^{-1} \quad (47)$$

is Schur stable. Then, by using [9, Lemma 2] we have that $\bar{\mathcal{A}} = (W^T \otimes I)\mathcal{M}$ is Schur stable, because W satisfies the conditions in (15).

Since the assumptions of Proposition 4.2 hold, then $\mathcal{F}_t \rightarrow \bar{\mathcal{F}}$ and $\mathcal{G}_t \rightarrow \bar{\mathcal{G}}$ over the interval $[\alpha T, \beta T]$ as $T \rightarrow \infty$. Moreover,

$$\begin{aligned} \bar{\mathcal{F}} &= \begin{bmatrix} \bar{\mathcal{A}} & \star \\ 0 & (A - \bar{G}C) + (\Gamma_B - \bar{G}\Gamma_D)\Gamma_{\bar{H}} \end{bmatrix} \\ \bar{\mathcal{G}} &= \begin{bmatrix} \bar{\mathcal{B}} \\ (\Gamma_B - \bar{G}\Gamma_D)\Gamma_{\bar{L}} \end{bmatrix} \end{aligned}$$

where $\bar{B} := -(W^T \otimes I)(I \otimes A)(\bar{V}^{-1} + \mathcal{S})^{-1}(J^T \otimes I)CR^{-1}D\bar{L} + \mathbf{1} \otimes B\bar{N}$ with $\Gamma_{L_t} \rightarrow \Gamma_L = [\bar{N}^T \ \bar{L}^T]^T$. Since $\bar{\mathcal{F}}$ is block upper-triangular, its eigenvalues coincides with the eigenvalues of \bar{A} and $(A - \bar{G}C) + (\Gamma_B - \bar{G}\Gamma_D)\Gamma_H$. Since the latter are Schur stable, we conclude that $\bar{\mathcal{F}}$ is Schur stable.

The Lyapunov equation in (35) satisfies the assumptions of Proposition 4.3 for c sufficiently small. Accordingly, \mathcal{P}_t converges to $\bar{\mathcal{P}}$ over $[\alpha T, \beta T]$ as $T \rightarrow \infty$ which is the unique solution to (44). \square

B. Optimality property under large deviations

We shall show that, under the least favorable model solution to (4) with c sufficiently large, it is very likely that the predictor at node k of Algorithm 1 performs better than the predictor at node k based on the scheme in [9]. The latter scheme, indeed, does not consider the possibility that the actual model and the nominal model do not coincide exactly.

Without loss of generality we assume that y_t is obtained by stacking first $y_{l,t}$ with $l \in \mathcal{N}_k$ and then $y_{l,t}$ with $l \notin \mathcal{N}_k$. Then, it is not difficult to see that that the standard local Kalman predictor at node k coincides with the Kalman predictor based on the model

$$\begin{aligned} x_{t+1} &= Ax_t + Bw_t \\ y_t &= \begin{bmatrix} C_k^{loc} \\ 0 \end{bmatrix} x_t + \begin{bmatrix} D_k^{loc} \\ \check{F}_{k,t}^{loc} \end{bmatrix} v_t \end{aligned} \quad (48)$$

where $\check{F}_{k,t}^{loc}$ is an arbitrary invertible matrix because the observations $y_{l,t}$ with $l \notin \mathcal{N}_k$ play no role at node k . Let $Q_{k,t}^{loc} := \check{F}_{k,t}^{loc}(\check{F}_{k,t}^{loc})^T$. Therefore the robust intermediate prediction at node k in Algorithm 1, hereafter denoted by RKF diff, is the solution of the mini-max problem

$$\hat{\psi}_{k,t+1} = \operatorname{argmin}_{g_t \in \mathbb{G}_{k,t}} \max_{\tilde{\phi}_{k,t} \in \mathbb{B}_{k,t}} \tilde{\mathbb{E}}_k[\|x_{t+1} - g_t\|^2 | Y_{t-1}] \quad (49)$$

where

$$\mathbb{B}_{k,t} := \{ \tilde{\phi}_{k,t} \text{ s.t. } \tilde{\mathbb{E}}_k[\log(\tilde{\phi}_{k,t}/\phi_{k,t}) | Y_{t-1}] \leq c \}; \quad (50)$$

$\phi_{k,t}$ is the transition probability density of z_t given x_t corresponding to (48) and $\tilde{\phi}_{k,t}$ is the least favorable one in $\mathbb{B}_{k,t}$; $\tilde{\mathbb{E}}_k[\log(\tilde{\phi}_{k,t}/\phi_{k,t}) | Y_{t-1}]$ and $\tilde{\mathbb{E}}_k[\|x_{t+1} - g_t\|^2 | Y_{t-1}]$ are defined as in (3) and (5), respectively, with ϕ_t , $\tilde{\phi}_t$ and \tilde{f}_t replaced by $\phi_{k,t}$, $\tilde{\phi}_{k,t}$ and $\tilde{f}_{k,t}$, respectively; $\tilde{f}_{k,t}(x_t | Y_{t-1}) \sim \mathcal{N}(\hat{x}_{k,t}, V_{k,t})$ is the least favorable conditional probability density of x_t given Y_{t-1} at node k ; $\mathbb{G}_{k,t}$ denotes the set of all estimators such that $\tilde{\mathbb{E}}_k[\|g_t\|^2]$ is finite for any $\tilde{\phi}_{k,t} \in \mathbb{B}_{k,t}$.

In [18] it has been shown that the mini-max problem (49) can be reformulated in terms of

$$\bar{p}_t^{loc}(z_t | Y_{t-1}) = \int \phi_{k,t}(z_t | x_t) \tilde{f}_{k,t}(x_t | Y_{t-1}) dx_t \quad (51)$$

$$\tilde{p}_t^{loc}(z_t | Y_{t-1}) = \int \tilde{\phi}_{k,t}(z_t | x_t) \tilde{f}_{k,t}(x_t | Y_{t-1}) dx_t \quad (52)$$

representing the local pseudo-nominal and the local least favorable conditional probability density of z_t given Y_{t-1} .

Moreover, $\mathbb{D}_{KL}(\tilde{p}_t^{loc}, \bar{p}_t^{loc}) = c$ and

$$\mathbb{D}_{KL}(\tilde{p}_t^{loc}, \bar{p}_t^{loc}) = \int \tilde{p}_t^{loc}(z_t|Y_{t-1}) \log \left(\frac{\tilde{p}_t^{loc}(z_t|Y_{t-1})}{\bar{p}_t^{loc}(z_t|Y_{t-1})} \right) dz_t \quad (53)$$

is the Kullback-Leibler divergence between \tilde{p}_t^{loc} and \bar{p}_t^{loc} .

Remark 3: It is worth noting that $\mathbb{D}_{KL}(p_1, p_2)$ represent the negative log-likelihood (up to constant factors) of p_2 under the the model described by p_1 , [29]. Assume that $\mathbb{D}_{KL}(p_1, p_2) \ll \mathbb{D}_{KL}(p_1, p_3)$ where the symbol \ll means “much less than”. This means that p_2 explains the data generated by p_1 better than p_3 .

Notice that the best intermediate prediction at node k is the one constructed using the least favorable model

$$\tilde{p}_t(z_t|Y_{t-1}) = \int \tilde{\phi}_t(z_t|x_t) \tilde{f}_{k,t}(x_t|Y_{t-1}) dx_t \quad (54)$$

since we are evaluating the performance under the least favorable model solution to the “centralized” mini-max problem in (4) which is not available because it requires to compute the centralized filtering gains of the centralized robust Kalman filter. On the other hand, the intermediate prediction at node k of the algorithm proposed in [9], hereafter denoted by KF diff, is constructed using the nominal model

$$p_t^{loc}(z_t|Y_{t-1}) = \int \phi_{k,t}(z_t|x_t) f_{k,t}(x_t|Y_{t-1}) dx_t \quad (55)$$

where $f_{k,t}(x_t|Y_{t-1})$ is the nominal conditional probability density of x_t given Y_{t-1} at node k . In view of Remark 3, the next proposition shows that if c is sufficiently large, then \tilde{p}_t^{loc} explains the measurements generated by the actual model \tilde{p}_t better than p_t^{loc} . As a consequence, it is very likely that the performance of $\psi_{k,t+1}$ using RKF diff (i.e. using \tilde{p}_t^{loc}) is better than the one using KF diff (i.e. using p_t^{loc}).

Proposition 4.5: Assume that for some t the distribution of $x_{k,t}$ given Y_{t-1} at node k is fixed and it is the same for RKF diff, KF diff, that is $f_{k,t}(x_t|Y_{t-1})$ and $\tilde{f}_{k,t}(x_t|Y_{t-1})$ coincide. Then, for c sufficiently large we have that

$$\mathbb{D}_{KL}(\tilde{p}_t, \tilde{p}_t^{loc}) \ll \mathbb{D}_{KL}(\tilde{p}_t, p_t^{loc}). \quad (56)$$

Proof: Let $\tilde{f}_{k,t} \sim \mathcal{N}(\hat{x}_{k,t}, V_{k,t})$ with $V_{k,t} > 0$ which is fixed and thus it does not depend on c . First, notice that $p_t(z_t|Y_{t-1}) = \bar{p}_t^{loc}(z_t|Y_{t-1})$ because the distribution of $x_{k,t}$ given Y_{t-1} is the same for RKF diff and KF diff. Accordingly,

$$\begin{aligned} p_t^{loc}(z_t|Y_{t-1}) &\sim \mathcal{N}(\mu_t^{loc}, K_t^{loc}) \\ \tilde{p}_t^{loc}(z_t|Y_{t-1}) &\sim \mathcal{N}(\mu_t^{loc}, \tilde{K}_t^{loc}) \\ \tilde{p}_t(z_t|Y_{t-1}) &\sim \mathcal{N}(\mu_t, \tilde{K}_t) \end{aligned} \quad (57)$$

where

$$\mu_t^{loc} = \begin{bmatrix} A \\ C_k^{loc} \\ 0 \end{bmatrix} \hat{x}_{k,t}, \quad \mu_t = \begin{bmatrix} A \\ C_k^{loc} \\ \check{C}_k^{loc} \end{bmatrix} \hat{x}_{k,t}, \quad (58)$$

$$\begin{aligned}
K_t^{loc} &= \begin{bmatrix} A \\ C_k^{loc} \\ 0 \end{bmatrix} V_{k,t} \begin{bmatrix} A^T & (C_k^{loc})^T & 0 \end{bmatrix} + \begin{bmatrix} BB^T & 0 & 0 \\ 0 & R_k^{loc} & 0 \\ 0 & 0 & Q_{k,t}^{loc} \end{bmatrix}, \\
\tilde{K}_t^{loc} &= K_t^{loc} + \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} (V_{k,t+1} - P_{k,t+1}) \begin{bmatrix} I & 0 & 0 \end{bmatrix}, \\
K_t &= \begin{bmatrix} A \\ C_k^{loc} \\ \check{C}_k^{loc} \end{bmatrix} V_{k,t} \begin{bmatrix} A^T & (C_k^{loc})^T & (\check{C}_k^{loc})^T \end{bmatrix} + \begin{bmatrix} BB^T & 0 & 0 \\ 0 & R_k^{loc} & 0 \\ 0 & 0 & \check{R}_k^{loc} \end{bmatrix}, \\
\tilde{K}_t &= K_t + \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} (V_{t+1} - P_{t+1}) \begin{bmatrix} I & 0 & 0 \end{bmatrix};
\end{aligned}$$

\check{C}_k^{loc} and \check{R}_k^{loc} are the matrices obtained by using C_l and R_l , respectively, with $l \notin \mathcal{N}_k$. It is worth noting that the relation between \tilde{K}_t^{loc} and K_t^{loc} is given by [18, Theorem 1]. The same observation holds between \tilde{K}_t and K_t where the latter represents the covariance matrix of z_t given Y_{t-1} in the nominal model. Moreover,

$$\begin{aligned}
P_{k,t+1} &= AV_{k,t}A^T - AV_{k,t}(C_k^{loc})^T \left(C_k^{loc}V_{k,t}(C_k^{loc})^T + R_k^{loc} \right)^{-1} C_k^{loc}V_{k,t}A^T + BB^T \\
V_{k,t+1} &= (P_{k,t+1}^{-1} - \theta_{k,t}I)^{-1} \\
P_{t+1} &= AV_{k,t}A^T - AV_{k,t} \begin{bmatrix} (C_k^{loc})^T & (\check{C}_k^{loc})^T \end{bmatrix} \left(\begin{bmatrix} C_k^{loc} \\ \check{C}_k^{loc} \end{bmatrix} V_{k,t} \begin{bmatrix} (C_k^{loc})^T & (\check{C}_k^{loc})^T \end{bmatrix} + \begin{bmatrix} R_k^{loc} & 0 \\ 0 & \check{R}_k^{loc} \end{bmatrix} \right)^{-1} \begin{bmatrix} C_k^{loc} \\ \check{C}_k^{loc} \end{bmatrix} V_{k,t}A^T + BB^T \\
V_{t+1} &= (P_{t+1}^{-1} - \theta_t I)^{-1}
\end{aligned}$$

and $\theta_{k,t}$, θ_t are the solution to $\gamma(P_{k,t+1}, \theta_{k,t}) = c$, $\gamma(P_{t+1}, \theta_t) = c$, respectively. Recall that

$$\gamma(P, \theta) := \log \det(I - \theta P) + \text{tr}((I - \theta P)^{-1} - I). \quad (59)$$

In view of (57), it is not difficult to see that

$$\mathbb{D}_{KL}(\tilde{p}_t, \tilde{p}_t^{loc}) = \mathbb{D}_{KL}(\tilde{p}_t, p_t^{loc}) + \frac{1}{2}d_\Delta \quad (60)$$

where

$$\begin{aligned}
d_\Delta &= \delta^T ((\tilde{K}_t^{loc})^{-1} - (K_t^{loc})^{-1}) \delta + \log \det(\tilde{K}_t^{loc}) \\
&\quad - \text{tr}(\tilde{K}_t (K_t^{loc})^{-1}) + \text{tr}(\tilde{K}_t (\tilde{K}_t^{loc})^{-1}) - \log \det(K_t^{loc}) \\
&\leq \log \det(\tilde{K}_t^{loc}) + \text{tr} \left[\tilde{K}_t \left((\tilde{K}_t^{loc})^{-1} - (K_t^{loc})^{-1} \right) \right] - \log \det(K_t^{loc})
\end{aligned} \quad (61)$$

where $\delta = \mu_t - \mu_t^{loc}$ and we have exploited the fact that $(\tilde{K}_t^{loc})^{-1} - (K_t^{loc})^{-1} \leq 0$ because $P_{k,t+1} < V_{k,t+1}$ and thus $\tilde{K}_t^{loc} \geq K_t^{loc}$. Moreover, after some algebraic manipulations we obtain

$$d_\Delta \leq n \log \|V_{k,t+1}\| - \beta_{k,t} \|V_{t+1}\| + \nu_{k,t} \quad (62)$$

where

$$\begin{aligned}
\beta_{k,t} &= \lambda_{\min}(P_{k,t+1}^{-1} [P_{k,t+1}^{-1} + (V_{k,t+1} - P_{k,t+1})^{-1}]^{-1} P_{k,t+1}^{-1})^{-1} \text{tr}(\tilde{V}_{t+1} - \|V_{t+1}\|^{-1} P_{t+1}) \\
\nu_{k,t} &= -\log \det K_t^{loc} + (Np + n) \log \lambda_{\max}(K_t^{loc}) + \log \det(\|V_{k,t+1}\|^{-1} I_n + \lambda_{\max}(K_t^{loc})^{-1} \tilde{V}_{k,t+1})
\end{aligned}$$

$\lambda_{max}(K_t^{loc})$ denotes the maximum eigenvalue of K_t^{loc} , $\bar{V}_{k,t+1} := \|V_{k,t+1}\|^{-1}V_{k,t+1}$ and $\bar{V}_{t+1} := \|V_{t+1}\|^{-1}V_{t+1}$.

It [17] it has been shown that the mapping $c \mapsto \|V_{k+1,t}\|$ has singular value which is positive. Accordingly, if we take a sequence $c^{(m)}$, $m \in \mathbb{N}$, such that $c^{(m)} > 0$ and $c^{(m)} \rightarrow \infty$ as $m \rightarrow \infty$, then $\|V_{k,t+1}^{(m)}\| \rightarrow \infty$. The same reasoning holds for the mapping $c \mapsto \|V_{t+1}\|$ and thus $\|V_{t+1}^{(m)}\| \rightarrow \infty$. Consider the sequences $\bar{V}_{k,t+1}^{(m)} := \|V_{k,t+1}^{(m)}\|^{-1}V_{k,t+1}^{(m)}$ and $\bar{V}_{t+1}^{(m)} := \|V_{t+1}^{(m)}\|^{-1}V_{t+1}^{(m)}$ which belong to the compact set $\mathcal{U} := \{V \text{ s.t. } \|V\| = 1\}$. Therefore, there exist the subsequences $\bar{V}_{k,t+1}^{(m_l)}$, $l \in \mathbb{N}$ and $\bar{V}_{t+1}^{(m_l)}$, $l \in \mathbb{N}$, converging to $\bar{V}_{k,t+1}^{(\infty)}$ and $\bar{V}_{t+1}^{(\infty)}$, respectively. It is worth noting that $\bar{V}_{k,t+1}^{(\infty)}, \bar{V}_{t+1}^{(\infty)} \geq 0$ and different from the null matrix because $\bar{V}_{k,t+1}^{(\infty)}, \bar{V}_{t+1}^{(\infty)} \in \mathcal{U}$. Accordingly, if we consider the corresponding subsequences for $\beta_{k,t}$ and $\nu_{k,t}$, we have: $\beta_{k,t}^{(m_l)} \rightarrow \lambda_{min}(P_{k,t+1}^{-1})^{-1} \text{tr}(\bar{V}_{t+1}) > 0$ and $\nu_{k,t}^{(m_l)}$ is bounded above.

Next we show that $\|V_{t+1}^{(m_l)}\|/\|V_{k,t+1}^{(m_l)}\| \rightarrow \zeta > 0$. First, we recall that $V_{k,t+1}^{(m_l)}$ and $V_{t+1}^{(m_l)}$ are given by $\theta_{k,t}^{(m_l)}$ and $\theta_t^{(m_l)}$, respectively. In particular, we have $\gamma(P_{t+1}^{(m_l)}, \theta_t^{(m_l)}) = c^{(m_l)}$. Notice that we can rewrite the latter as

$$\sum_{i=1}^n \log(1 - d_i \theta_t^{(m_l)}) + (1 - \theta_t^{(m_l)})^{-1} - 1 = c^{(m_l)} \quad (63)$$

where $d_i \geq d_{i+1}$ denotes the eigenvalues of P_{t+1} and $0 < \theta_t^{(m_l)} < d_1^{-1}$. In what follows we assume that the eigenvalue d_1 has multiplicity equal to one, and thus $d_1 > d_i$ with $i \geq 2$. This assumption is not restrictive, indeed it generically holds. Then we can rewrite (63) as

$$f(d_1 \theta_t^{(m_l)}) + \check{c}^{(m_l)} = c^{(m_l)}$$

where

$$\begin{aligned} f(x) &= \log(1 - x) + (1 - x)^{-1} - 1 \\ \check{c}^{(m_l)} &= \sum_{i=2}^n \log(1 - d_i \theta_t^{(m_l)}) + (1 - \theta_t^{(m_l)})^{-1} - 1, \end{aligned}$$

$\check{c}^{(m_l)} \rightarrow \check{c}$ and \check{c} is a bounded value. Therefore

$$f(d_1 \theta_t^{(m_l)}) = c^{(m_l)} - \check{c}^{(m_l)}.$$

Since $c^{(m_l)} \rightarrow \infty$, we have $\check{c}^{(m_l)} = o(c^{(m_l)})$, i.e. $\check{c}^{(m_l)}/c^{(m_l)} \rightarrow 0$ as l tends to infinity. Accordingly,

$$f(d_1 \theta_t^{(m_l)}) = c^{(m_l)} - o(c^{(m_l)}). \quad (64)$$

The same reasoning applies for $\theta_{k,t}^{(m_l)}$:

$$f(d_{k,1} \theta_{k,t}^{(m_l)}) = c^{(m_l)} - o(c^{(m_l)}) \quad (65)$$

where $d_{k,i} \geq d_{k,i+1}$ are the eigenvalues of $P_{k,t+1}$ and $d_{k,1}$ has multiplicity equal to one. Notice that $d_1 \theta_t^{(m_l)}$ and $d_{k,1} \theta_{k,t}^{(m_l)}$ belong to the interval $[0, 1)$. It is not difficult to see that $f : [0, 1) \rightarrow [0, \infty)$ is monotone increasing in the interval $[0, 1)$. Accordingly, it admits the continuous inverse function $g : [0, \infty) \rightarrow [0, 1)$ and

$$\begin{aligned} \theta_t^{(m_l)} &= d_1^{-1} g(c^{(m_l)} - o(c^{(m_l)})) \\ \theta_{k,t}^{(m_l)} &= d_{k,1}^{-1} g(c^{(m_l)} - o(c^{(m_l)})). \end{aligned}$$

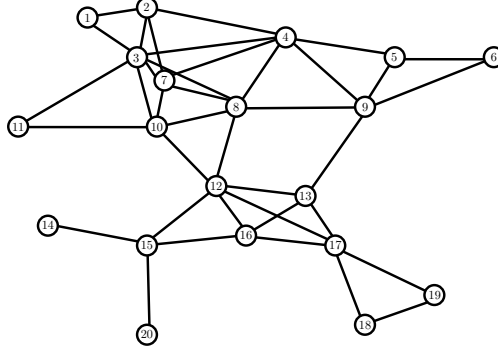


Fig. 1. Network of 20 sensors for collecting noisy position measurements of the projectile.

Notice that

$$\begin{aligned} \lim_{l \rightarrow \infty} g \left(c^{(m_l)} - o(c^{(m_l)}) \right) &= g \left(\lim_{l \rightarrow \infty} c^{(m_l)} - o(c^{(m_l)}) \right) \\ &= g \left(\lim_{l \rightarrow \infty} c^{(m_l)} \lim_{l \rightarrow \infty} \left(1 - \frac{o(c^{(m_l)})}{c^{(m_l)}} \right) \right) = g \left(\lim_{l \rightarrow \infty} c^{(m_l)} \right) = \lim_{l \rightarrow \infty} g \left(c^{(m_l)} \right) \end{aligned} \quad (66)$$

Finally, we have

$$\begin{aligned} \lim_{l \rightarrow \infty} \frac{\|V_{t+1}^{(m_l)}\|}{\|V_{k,t+1}^{(m_l)}\|} &= \lim_{l \rightarrow \infty} \sqrt{\frac{\sum_{i=1}^n \frac{1}{d_i^{-1} - \theta_t^{(m_l)}}}{\sum_{i=1}^n \frac{1}{d_{k,i}^{-1} - \theta_{k,t}^{(m_l)}}}} = \lim_{l \rightarrow \infty} \sqrt{\frac{\frac{1}{d_1^{-1} - \theta_t^{(m_l)}} + \sum_{i=2}^n \frac{1}{d_i^{-1} - \theta_t^{(m_l)}}}{\frac{1}{d_{k,1}^{-1} - \theta_{k,t}^{(m_l)}} + \sum_{i=2}^n \frac{1}{d_{k,i}^{-1} - \theta_{k,t}^{(m_l)}}}} \\ &= \lim_{l \rightarrow \infty} \sqrt{\frac{\frac{d_1}{1 - g(c^{(m_l)} - o(c^{(m_l)}))} + \sum_{i=2}^n \frac{1}{d_i^{-1} - \theta_t^{(m_l)}}}{\frac{d_{k,1}}{1 - g(c^{(m_l)} - o(c^{(m_l)}))} + \sum_{i=2}^n \frac{1}{d_{k,i}^{-1} - \theta_{k,t}^{(m_l)}}}} = \lim_{l \rightarrow \infty} \sqrt{\frac{\frac{d_1}{1 - g(c^{(m_l)})} + \sum_{i=2}^n \frac{1}{d_i^{-1} - \theta_t^{(m_l)}}}{\frac{d_{k,1}}{1 - g(c^{(m_l)})} + \sum_{i=2}^n \frac{1}{d_{k,i}^{-1} - \theta_{k,t}^{(m_l)}}}} \\ &= \lim_{l \rightarrow \infty} \sqrt{\frac{\frac{d_1}{1 - g(c^{(m_l)})}}{\frac{d_{k,1}}{1 - g(c^{(m_l)})}}} \end{aligned} \quad (67)$$

where we exploited the fact that $\lim_{x \rightarrow \infty} g(x) = 1$ in the last equality. Then, we have

$$\lim_{l \rightarrow \infty} \frac{\|V_{t+1}^{(m_l)}\|}{\|V_{k,t+1}^{(m_l)}\|} = \lim_{l \rightarrow \infty} \sqrt{\frac{\frac{d_1}{1 - g(c^{(m_l)})}}{\frac{d_{k,1}}{1 - g(c^{(m_l)})}}} = \sqrt{\frac{d_1}{d_{k,1}}} > 0. \quad (68)$$

Accordingly the corresponding subsequence $d_{\Delta}^{(m_l)}$ approaches $-\infty$ because the term $-\beta_{k,t}^{(m_l)} \|V_{t+1}^{(m_l)}\|$ dominates the logarithmic term $n \log \|V_{k,t+1}^{(m_l)}\|$. We conclude that for c sufficiently large (56) holds. \square

V. NUMERICAL EXAMPLE

In order to evaluate the performance of the distributed robust Kalman filters, we consider the problem in [9] of tracking the position of a projectile by using noisy position measurements obtained by a network of $N = 20$ sensors depicted in Figure 1. The model for the projectile motion is

$$\dot{x}_t^c = \Phi x_t^c + u_t^c \quad (69)$$

where

$$\Phi = \begin{bmatrix} 0 & 0 \\ I_3 & 0 \end{bmatrix},$$

$u_t^c = [0 \ 0 \ -g \ 0 \ 0 \ 0]^T$, with $g = -10$, and $x_t^c = [v_{x,t} \ v_{y,t} \ v_{z,t} \ p_{x,t} \ p_{y,t} \ p_{z,t}]^T$ with v denoting the velocity, p the position and the subscripts x, y, z denoting the three spatial dimensions. We discretize (69) by using a sampling time equal to 0.1. In this way we obtain the discrete time model $x_{t+1} = Ax_t + u_t$ where x_t is the sampled version of x_t^c , $A = I_6 + 0.1\Phi$ and $u_t = (0.1I_6 + 0.1^2\Phi/2)u_t^c$. We assume that every sensor measures the position of the projectile in either two horizontal dimensions, or a combination of one horizontal dimension and the vertical dimension (i.e. one sensor does not have measurements in all the three dimensions). Therefore, we obtain the nominal discrete state-space model (9) where $C_k = [0 \ 0 \ 0 \ \text{diag}(1, 1, 0)]$, in the case that the sensor measures only the horizontal positions, or $C_k = [0 \ 0 \ 0 \ \text{diag}(1, 0, 1)]$, $C_k = [0 \ 0 \ 0 \ \text{diag}(0, 1, 1)]$, in the case that the sensor measures one horizontal position and the vertical position. Moreover, we choose $B = \sqrt{0.001}I$, $R_k = D_k D_k^T = \sqrt{k}PR_0P^T$ where $R_0 = 0.5 \cdot \text{diag}(1, 4, 7)$ and P is a permutation matrix randomly chosen for every node. Finally, the initial state x_0 is a Gaussian random vector with covariance matrix $P_0 = I$.

In what follows, we consider the following predictors:

- **RKF diff** – the distributed robust Kalman filter with diffusion step in Algorithm 1; the diffusion matrix W is chosen as

$$w_{lk} = \begin{cases} \alpha_k n_l, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise,} \end{cases} \quad (70)$$

where n_l denotes the number of neighbors of node l and $\alpha_k > 0$ is a normalization parameter chosen in such a way that (15) holds.

- **KF diff** – the distributed Kalman filter with diffusion step proposed in [9]; the diffusion matrix W is chosen as in (70).
- **RKF cons** – the distributed robust Kalman filter in Algorithm 1 with the consensus-based update in (17); the consensus parameter is set equal to $\varepsilon = 0.1$.
- **KF cons** – the distributed Kalman filter with consensus-based update proposed in [7]; the consensus parameter is set equal to $\varepsilon = 0.1$.
- **RKF local** – the local robust Kalman filter in Algorithm 1 with $W = I$.
- **KF local** – the local Kalman filter proposed in [24, p. 329].
- **RKF central** – the centralized robust Kalman filter proposed [18].
- **KF central** – the centralized Kalman filter.

In the first experiment we assume that the actual state-space model belongs to the ball defined in (2) about the aforementioned nominal model and with tolerance $c = 0.02$. The average least favorable mean square deviation across the network is depicted in Figure 2. As we can see, \overline{MSD}_t converges in steady state for any algorithm. The local algorithms **RKF local** and **KF local** provide the worst performance and the robust version behaves slightly better than the standard version in steady state. The consensus-based algorithms **RKF cons** and **KF cons**

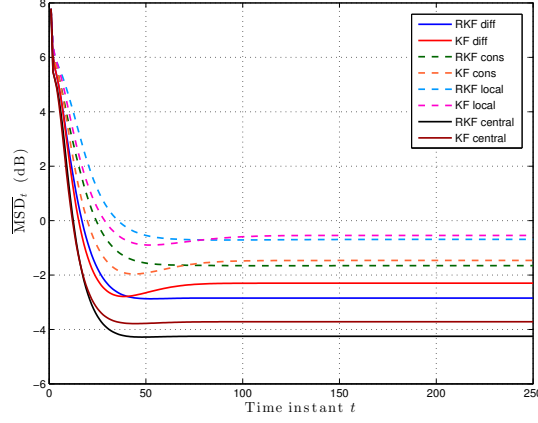


Fig. 2. Least favorable mean square deviation across the network with tolerance $c = 0.02$.

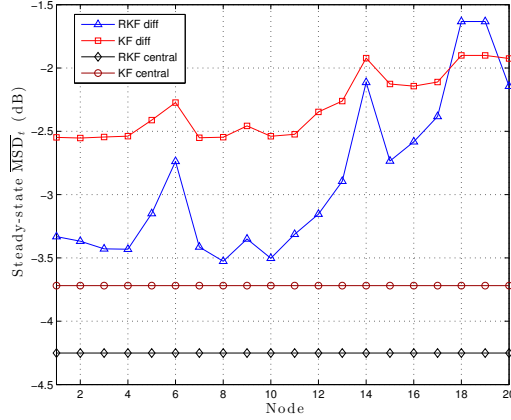


Fig. 3. Least favorable mean square deviation for each node in steady state with tolerance $c = 0.02$.

perform better than the latter and the robust version behaves slightly better than the standard one in steady state. The diffusion-based algorithms RKF diff and KF diff provides the best distributed performance, in particular RKF diff performs better than KF diff. Finally, the centralized algorithm provides the best performance and RKF is the best predictor. The least favorable mean square deviation for each node in steady state for the diffusion-based and centralized algorithms is depicted in Figure 3. As we can see, RKF diff provides a better performance than KF diff in the majority of the nodes. Finally, Figure 4 shows the risk sensitivity parameters $\theta_{k,t}$ of RKF diff and the risk sensitivity parameter θ_t of RKF. We notice that the former are less than the latter. Therefore, RKF diff reduces the risk sensitivity parameters over the network in respect to RKF. Such a reduction can be justified as follows. First, the larger the risk sensitivity parameter is, the more large errors are penalized, as noticed in [18]. Then, it is worth observing that RKF cons and RKF local have the same risk sensitivity parameters of RKF diff, indeed the value

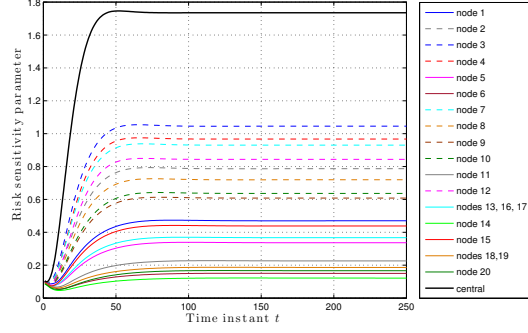


Fig. 4. Risk sensitivity parameters $\theta_{k,t}$, $k = 1 \dots 20$, of RKF diff and the risk sensitivity parameter of the centralized filter RKF (black line) with $c = 0.02$.

of $\theta_{k,t}$ does not depend on the matrix W . So, without loss of generality, we can consider RKF local. RKF local at node k and RKF are the same algorithm, but applied on a different state space model. The state space model used for RKF local at node k is characterized by a subset of observations of the state space model used for RKF. Since the mismatch modeling budget c is the same for both the models, then it means that the observations of the least favorable model of RKF are affected by more uncertainty than the ones of RKF local. Accordingly, it is required to penalize large errors in RKF more severely than in RKF local, hence θ_t must be greater than $\theta_{k,t}$.

In the second experiment we have considered the case that in the actual model there are large deviations in respect to the nominal one. More precisely, we have chosen $c = 0.06$. The least favorable mean square deviation across the network is depicted in Figure 5. As we can see, all the robust distributed algorithms outperform the corresponding standard distributed algorithms in steady state. Among the robust algorithms, RKF diff gives the best performance, then we have RKF cons and finally RKF local. The least favorable mean square deviation for each node in steady state for the diffusion-based and centralized algorithms is depicted in Figure 6. As we can see, RKF diff provides a better performance than KF diff in almost all nodes: the unique exception regards two nodes wherein RKF diff performs slightly worse than KF diff. The risk sensitivity parameters $\theta_{k,t}$ of RKF diff and the risk sensitivity parameter θ_t of RKF are depicted in Figure 7. Also in this case, RKF diff reduces the risk sensitivity parameters over the network in respect to RKF. On the hand, the values of all these risk sensitivity parameters has been increased in respect to the case $c = 0.02$. Indeed, in the current case the mismatch modeling budget has been increased and thus it is required to penalize large error more severely.

VI. CONCLUSIONS

In this paper, we have considered a filtering problem over a sensor network and under model uncertainty. We have proposed a robust distributed algorithm with diffusion step. We have derived the least favorable performance for this algorithm and showed that the least favorable mean square deviation across the network does converge to a finite constant value provided that the mismatch modeling budget allowed for each time step is sufficiently small.

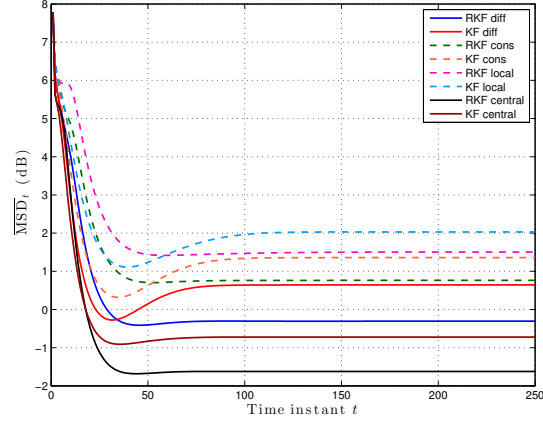


Fig. 5. Least favorable mean square deviation across the network with tolerance $c = 0.06$.

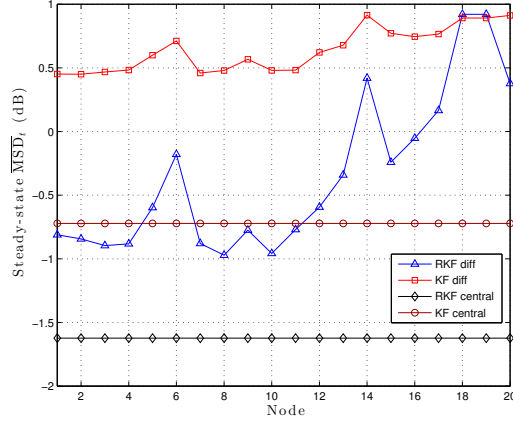


Fig. 6. Least favorable mean square deviation for each node in steady state with tolerance $c = 0.06$.

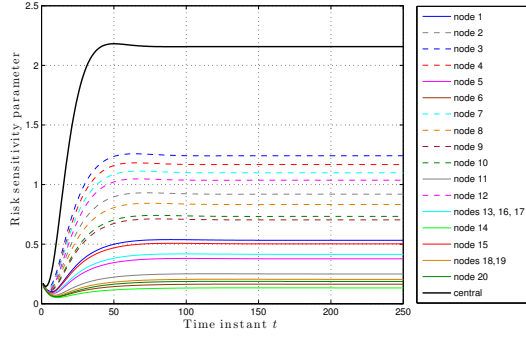


Fig. 7. Risk sensitivity parameters $\theta_{k,t}$, $k = 1 \dots 20$, of RKF diff and the risk sensitivity parameter of the centralized filter RKF (black line) with $c = 0.06$.

Finally, a numerical example showed that this robust algorithm is preferable than the standard one in the presence of model deviations.

REFERENCES

- [1] H. N. Wu and H. D. Wang, “Distributed consensus observers-based H_∞ control of dissipative pde systems using sensor networks,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 112–121, 2015.
- [2] L. Orihuela, P. Milln, C. Vivas, and F. R. Rubio, “Distributed control and estimation scheme with applications to process control,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1563–1570, 2015.
- [3] T. Sadamoto, T. Ishizaki, and J. Imura, “Average state observers for large-scale network systems,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 761–769, 2017.
- [4] S. Wang, W. Ren, and J. Chen, “Fully distributed dynamic state estimation with uncertain process models,” *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2017.
- [5] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Approximate distributed Kalman filtering in sensor networks with quantifiable performance,” in *Fourth International Symposium on Information Processing in Sensor Networks*, April 2005, pp. 133–139.
- [6] R. Olfati-Saber, “Distributed Kalman filter with embedded consensus filters,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 8179–8184.
- [7] —, “Distributed Kalman filtering for sensor networks,” in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.
- [8] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed Kalman filtering based on consensus strategies,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.
- [9] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.
- [10] S. Yang, T. Huang, J. Guan, Y. Xiong, and M. Wang, “Diffusion strategies for distributed Kalman filter with dynamic topologies in virtualized sensor networks,” *Mobile Information Systems*, 2016.
- [11] E. Song, Y. Zhu, J. Zhou, and Z. You, “Optimal Kalman filtering fusion with cross-correlated sensor noises,” *Automatica*, vol. 43, no. 8, pp. 1450–1456, 2007.
- [12] J. Xu, E. Song, Y. Luo, and Y. Zhu, “Optimal distributed Kalman filtering fusion algorithm without invertibility of estimation error and sensor noise covariances,” *IEEE Signal Processing Letters*, vol. 19, no. 1, pp. 55–58, 2012.
- [13] S.-L. Sun and Z.-L. Deng, “Multi-sensor optimal information fusion Kalman filter,” *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.
- [14] J. Feng and M. Zeng, “Optimal distributed Kalman filtering fusion for a linear dynamic system with cross-correlated noises,” *International Journal of Systems Science*, vol. 43, no. 2, pp. 385–398, 2012.
- [15] R. Boel, M. James, and I. Petersen, “Robustness and risk-sensitive filtering,” *IEEE Trans. Automat. Control*, vol. 47, no. 3, pp. 451–461, 2002.
- [16] L. Hansen and T. Sargent, *Robustness*. Princeton, NJ: Princeton University Press, 2008.
- [17] B. Levy and R. Nikoukhah, “Robust least-squares estimation with a relative entropy constraint,” *Information Theory, IEEE Transactions on*, vol. 50, no. 1, pp. 89–104, Jan. 2004.
- [18] —, “Robust state-space filtering under incremental model perturbations subject to a relative entropy tolerance,” *IEEE Trans. Automat. Control*, vol. 58, pp. 682–695, Mar. 2013.
- [19] M. Zorzi, “On the robustness of the Bayes and Wiener estimators under model uncertainty,” *Automatica*, vol. 83, pp. 133–140, 2017.
- [20] —, “Robust Kalman filtering under model perturbations,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, June 2017.
- [21] B. Shen, Z. Wang, and Y. Hung, “Distributed H_∞ -consensus filtering in sensor networks with multiple missing measurements: The finite-horizon case,” *Automatica*, vol. 46, no. 10, pp. 1682–1688, 2010.
- [22] Y. Luo, Y. Zhu, D. Luo, J. Zhou, E. Song, and D. Wang, “Globally optimal multisensor distributed random parameter matrices Kalman filtering fusion with applications,” *Sensors*, vol. 8, no. 12, pp. 8086–8103, 2008.
- [23] B. C. Levy and M. Zorzi, “A contraction analysis of the convergence of risk-sensitive filters,” *SIAM Journal on Control and Optimization*, vol. 54, no. 4, pp. 2154–2173, 2016.
- [24] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall, 2000.
- [25] A. Zenere and M. Zorzi, “On the coupling of model predictive control and robust Kalman filtering,” *IET Control Theory Applications*, vol. 12, no. 13, pp. 1873–1881, 2018.

- [26] M. Zorzi and B. C. Levy, “Robust Kalman filtering: Asymptotic analysis of the least favorable model,” in *57th IEEE Conference on Decision and Control (CDC)*, Dec 2018.
- [27] —, “On the convergence of a risk sensitive like filter,” in *54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 4990–4995.
- [28] M. Zorzi, “Convergence analysis of a family of robust Kalman filters based on the contraction principle,” *SIAM Journal on Control and Optimization*, vol. 55, no. 5, pp. 3116–3131, 2017.
- [29] C. Robert, “Machine learning, a probabilistic perspective,” 2014.