Robust Distributed Kalman Filtering: On the Choice of the Local Tolerance

Alessandro Emanuele®, Francesco Gasparotto, Giacomo Guerra and Mattia Zorzi *

Department of Information Engineering, University of Padova, Via Gradenigo 6/B, 35131 Padova, Italy; alessandro.emanuele@studenti.unipd.it (A.E.); francesco.gasparotto@studenti.unipd.it (F.G.); giacomo.guerra.2@studenti.unipd.it (G.G.)

* Correspondence: zorzimat@dei.unipd.it

Received: 23 April 2020; Accepted: 4 June 2020; Published: 7 June 2020

Abstract: We propose a distributed Kalman filter for a sensor network under model uncertainty. The distributed scheme is characterized by two communication stages in each time step: in the first stage, the local units exchange their observations and then they can compute their local estimate; in the final stage, the local units exchange their local estimate and compute the final estimate using a diffusion scheme. Each local estimate is computed in order to be optimal according to the least favorable model belonging to a prescribed local ambiguity set. The latter is a ball, in the Kullback–Liebler topology, about the corresponding nominal local model. We propose a strategy to compute the radius, called local tolerance, for each local ambiguity set in the sensor network, rather than keep it constant across the network. Finally, some numerical examples show the effectiveness of the proposed scheme.

Keywords: distributed robust Kalman filtering; least favorable analysis; sensor networks

1. Introduction

Modern problems involve a large number of sensors forming a sensor network and taking measurements from which we would like to infer quantities not accessible to observation possibly at each node location. These problems can be classified as filtering problems whose solution is given by the Kalman filter. On the other hand, its implementation is very expensive in terms of data transmission; indeed, we require that all sensors can exchange their measurements. Such a limitation disappears by considering distributed filtering [1–9]. The key idea is that the communication among the nodes is limited.

In the simplest distributed strategy, the state estimate of a node (i.e., local unit) is computed by using only the observations from its neighbors. Such a strategy, however, appears to be not very effective. A remarkable progress has been reached by distributed Kalman filtering with consensus [10–13] and diffusion strategies [14,15]. These distributed approaches are characterized by several communication stages during each time step. For instance, in the first stage, the local units exchange their observations and then they can compute their local estimate; in the final stage, the local units exchange their local estimate and compute the final estimate using a consensus or a diffusion scheme. Many important challenges have been addressed in distributed filtering. For instance, the issues about limited observability, network topologies that restrict allowable communications, and communication noises between sensors are considered in [16]; the case in which the sensor network is subject to transmission delays is considered in [17]; and the cases of missing measurements and absence of communication among the nodes are analyzed in [18,19], respectively. It is worth observing that distributed state estimation can be performed also through different principles [20–25].
For instance, each node can transmit its measurements to a fusion center and then the latter computes the state estimate.

An important aspect in filtering applications is that the nominal model does not correspond to the actual one. Risk sensitive Kalman filtering [26–28] addresses this problem by penalizing large errors. The severity of this penalization is tuned by the so-called risk sensitivity parameter: the larger the risk sensitivity parameter is, the more large errors are penalized. A refinement of these filters is given by robust Kalman filtering where the uncertainty is expressed incrementally [29–32]. More precisely, for each time step, the state estimator minimizes the prediction error according to the least favorable model belonging to a prescribed ambiguity set. The latter is a ball in the Kullback–Leibler (KL) topology whose center is the nominal model. The radius of this ball is called tolerance and it represents the discrepancy budget between the actual and the nominal model allowed for the corresponding time step. It is worth noting that the ambiguity set can be formed also by using different types of divergence (see, for instance, [33–35]).

The problem of distributed Kalman filtering under model uncertainty has been considered as well (see, e.g., [36–38]). In the present paper, we consider the distributed robust Kalman filter with diffusion step proposed in [39]. Here, the local estimate of each node is computed by using the robust Kalman filter in [29]. In this scenario, the least favorable model is the one used to compute the robust Kalman filter of the global model. Accordingly, we have one ambiguity set corresponding to the global model, which contains the actual model, and the local ambiguity sets corresponding to the nodes of the network. The centers of those balls are known because they are given by the nominal model. The local tolerances corresponding to the local ambiguity sets of the nodes are set equal to the one of the global ambiguity set. On the other hand, the local ambiguity set of a node corresponds to a local model which is just a part of the global model. Accordingly, taking all the local tolerances uniform across the network and equal to the global one may not be the best choice.

The main contribution of this paper is to propose a robust distributed Kalman filter as in [39] where the local tolerance for each node is customized. In this way, the local tolerance is non-uniform and time-varying across the network. We show through some simulation studies that the performance of the predictions is improved. Moreover, we show that, if the tolerance corresponding to the global ambiguity set is sufficiently small, then the local tolerances across the network are constant in the steady-state condition. Accordingly, it is also possible to simplify the distributed scheme by replacing the time-varying tolerances with the steady-state values.

The organization of this paper is as follows. In Section 2, we provide the background about robust Kalman filtering whose uncertainty is expressed incrementally. In Section 3, the distributed robust Kalman filter with local uniform tolerance is reviewed. In Section 4, we introduce the distributed robust Kalman filter with non-uniform local tolerance. In Section 5, we perform some numerical experiments to check the performance of the proposed distributed scheme. In Section 6, we propose an efficient approximation of the distributed robust Kalman filter with non-uniform local tolerance. Finally, in Section 7, we draw the conclusions.

Notation: \([a_{ij}]_{ij}\) denotes the matrix having entry \(a_{ij}\) in position \((i, j)\); \(A^T\) is the transposition of matrix \(A\); and \(A > 0 (A \geq 0)\) means that matrix \(A\) is positive (semi)-definite. \(\text{diag}(A_1 \ldots A_n)\) denotes a block-diagonal matrix whose blocks in the main block diagonal are \(A_1 \ldots A_n\). Given a squared matrix \(A\), \(\text{tr}(A)\) and \(|A|\) denote the trace and the determinant of \(A\), respectively. Given two matrices \(A\) and \(B\), \(A \otimes B\) denotes their Kronecker product. \(x \sim \mathcal{N}(m, K)\) means that \(x\) is a Gaussian random vector with mean \(m\) and covariance matrix \(K\).
2. Background

In this section, we review the robust Kalman filter proposed in [29], which represents the “building block” used throughout the paper. Consider the nominal state-space model

\[
x_{t+1} = Ax_t + \Gamma_B u_t + r_t \\
y_t = Cx_t + \Gamma_D u_t
\]

where \( A \in \mathbb{R}^{n \times n}, \Gamma_B \in \mathbb{R}^{n \times n+pN}, C \in \mathbb{R}^{pN \times n}, \Gamma_D \in \mathbb{R}^{pN \times n+pN}, \) \( t = \) the state process, \( y_t \) is the observation process, \( u_t \) is normalized white Gaussian noise (WGN), and \( r_t \) is a deterministic signal. It is assumed that \( u_t \) is independent from the initial state \( x_0 \sim \mathcal{N}(\hat{x}_0, V_0) \). We also assume that the noise entering in the state process and the one entering in the observation process are independent, i.e., we assume that \( \Gamma_B \Gamma_B^T = 0 \). Finally, the state-space model in Equation (1) is considered to be reachable and observable. Let \( \phi_t(z_t|x_t) \) denote the nominal transition probability density of \( z_t := [x_{t+1}^T y_t^T]^T \) given \( x_t \). Notice that \( \phi_t(z_t|x_t) \) is Gaussian by construction and it is straightforwardly given by Equation (1).

We assume that the (unknown) actual transition probability \( \tilde{\phi}_t(z_t|x_t) \) belongs to the ambiguity set which is a closed ball centered in \( \phi_t(z_t|x_t) \) in the KL topology:

\[
\mathbb{B}_t := \left\{ \phi_t \text{ s.t. } \mathbb{E} \left[ \log \left( \frac{\phi_t}{\phi_t} \right) \right] \leq c \right\}
\]

with

\[
\mathbb{E} \left[ \log \left( \frac{\tilde{\phi}_t}{\phi_t} \right) \right] := \int \int \tilde{\phi}_t(z_t|x_t) f_t(x_t|y_{t-1}) \log \left( \frac{\tilde{\phi}_t(z_t|x_t)}{\phi_t(z_t|x_t)} \right) dz_t dx_t,
\]

\( Y_{t-1} := \{ y_s, s = 0 \ldots t-1 \} \), and \( f_t(x_t|y_{t-1}) \sim \mathcal{N}(\hat{x}_t, V_t) \) is defined as the actual conditional probability density of \( x_t \) given \( y_{t-1} \). The mismatch modeling budget allowed for each time step is represented by the parameter \( c > 0 \), which is called tolerance. The robust estimator of \( x_{t+1} \) given \( y_t \) for the nominal model in Equation (1) is given by solving the following minimax problem:

\[
\hat{x}_{t+1} = \underset{\phi_t \in \mathbb{B}_t}{\arg\min\max_{g_t \in \mathcal{G}_t}} \mathbb{E} \left[ \| x_{t+1} - g_t(y_t) \|^2 \right]_{Y_{t-1}}
\]

where \( \mathcal{G}_t \) is the set of all estimators \( g_t \) whose variance is finite under any model in the ambiguity set \( \mathbb{B}_t \),

\[
\mathbb{E} \left[ \| x_{t+1} - g_t(y_t) \|^2 \right]_{Y_{t-1}} := \int \int \| x_{t+1} - g_t(y_t) \|^2 \tilde{\phi}_t(z_t|x_t) f_t(x_t|y_{t-1}) dz_t dx_t
\]

is the estimation error under the transition density \( \tilde{\phi}_t(z_t|x_t) \). In [29], it is proved that the estimator solution to the problem in Equation (3) has the following Kalman-like structure:

\[
G_t = AV_t C^T (CV_t C^T + \Gamma_D \Gamma_D^T)^{-1} \\
\hat{x}_{t+1} = Ax_t + G_t (y_t - C\hat{x}_t) + r_t \\
P_{t+1} = A \left( V_{t-1}^{-1} + C^T (\Gamma_D \Gamma_D^T)^{-1} C \right)^{-1} A^T + \Gamma_B \Gamma_B^T \\
\text{Find } \theta \text{ s.t. } \gamma(P_{t+1}, \theta_t) = c \\
V_{t+1} = (P_{t+1}^{-1} - \theta I)^{-1}
\]

where

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

Parameter \( \theta_t > 0 \) is called risk sensitivity parameter. It is worth noting that, given \( P > 0 \) and \( c > 0 \), the equation \( \gamma(P, \theta) = c \) always admits a unique solution in \( \theta \) and such that: \( \theta > 0, P^{-1} - \theta I > 0 \).
In the special case that \( c = 0 \), i.e., the nominal model coincides with the actual model, then \( \theta_t = 0 \) and thus Equation (4) degenerates in the usual Kalman filter.

**Remark 1.** It is worth noting that the robust Kalman filter is well defined also in the case that the ambiguity set Equation (2) is defined by a time-varying tolerance, i.e., \( c_t \) instead of \( c \). However, we prefer to keep \( c \) constant in Equation (3) because in the following we assume that the actual (global) model is the solution to Equation (3) with constant tolerance \( c \), in order to simplify the setup.

### 3. Distributed Robust Kalman Filtering with Uniform Local Tolerance

In this section, we review the distributed robust Kalman filter presented in [39]. Consider a network made by \( N \) sensors. The latter are connected if they can communicate with each other. Accordingly, every sensor \( k \) has a set of neighbors which is denoted by \( N_k \). In particular, \( k \in N_k \) that is each node is connected with itself. The number of neighbors of node \( k \) is denoted by \( n_k \).

The corresponding \( N \times N \) adjacency matrix \( J = [j_{lk}]_{N \times N} \) is defined as

\[
j_{lk} := \begin{cases} 1, & \text{if } l \in N_k \\ 0, & \text{otherwise.} \end{cases}
\]

We assume that every node collects a measurement \( y_{k,t} \in \mathbb{R}^p \) at time \( t \) and the corresponding nominal state-space model is

\[
x_{t+1} = Ax_t + Bw_t + r_t \\
y_{k,t} = C_k x_t + D_k v_{k,t} \quad k = 1 \ldots N
\]

where \( w_t \) and \( v_{k,t} \), with \( k = 1 \ldots N \), are independent normalized WGNs. It is worth noting that the actual state-space model for each node is unknown. By stacking Equation (5) for every \( k \), it is possible to rewrite such sensor network as Equation (1) where:

\[
y_t = \begin{bmatrix} y_{1,t} \\ \vdots \\ y_{N,t} \end{bmatrix}, \quad u_t = \begin{bmatrix} w_t \\ v_t \end{bmatrix}, \quad v_t = \begin{bmatrix} v_{1,t} \\ \vdots \\ v_{N,t} \end{bmatrix} \\
\Gamma_B = \begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix}, \quad \Gamma_D = \begin{bmatrix} 0 & D \end{bmatrix} \\
C = \begin{bmatrix} C_1 \\ \vdots \\ C_N \end{bmatrix}, \quad D = \text{diag}(D_1, \ldots, D_N).
\]

Accordingly, Equation (4) represents the centralized robust Kalman filter. Defining \( R := DD^T \), \( R_l := D_l D_l^T \) with \( l = 1 \ldots N \), and

\[
S_{\text{tot}} := C^T R^{-1} C = \sum_{l=1}^{N} C_l^T R_l^{-1} C_l,
\]

the Kalman gain for Equation (5) becomes, using the matrix inversion lemma,

\[
G_t = A \left( V_t^{-1} + S_{\text{tot}} \right)^{-1} C^T R^{-1}.
\]

Since the nominal model in Equation (5) does not coincide with the actual one and each node \( k \) can only exploit information shared by its neighbors \( l \in N_k \), the aim of distributed robust Kalman filtering is to compute a prediction \( \hat{x}_{k,t} \) of the state \( x_t \) for every node \( k \) by using only the local information,
taking into account the model uncertainty. In the case that the node \( k \) has access to all measurements across all the nodes in the network, then \( \hat{x}_{k,t} \) coincides with Equation (4) which can be written, using the parameterization in Equations (6) and (7) as

\[
\begin{align*}
\dot{x}_{k,t+1} &= A \dot{x}_{k,t} + A \left( V_{k,t}^{-1} + S_{tot} \right)^{-1} \sum_{l=1}^{N} C_l R_l^{-1} (y_{l,t} - C_l \dot{x}_{l,t}) + r_t \\
\dot{P}_{k,t+1} &= A \left( V_{k,t}^{-1} + S_{tot} \right)^{-1} A^T + BB^T \\
\text{Find } \theta_{k,t} \text{ s.t. } \gamma (P_{k,t+1}, \theta_{k,t}) = c \\
V_{k,t+1} &= \left( P_{k,t}^{-1} - \theta_{k,t} I \right)^{-1},
\end{align*}
\]  

(8)

where \( \dot{x}_{k,t} = \dot{x}_t, P_{k,t} = P_t, V_{k,t} = V_t, \) and \( \theta_{k,t} = \theta_t. \) In the case that not all the measurements in the network are accessible to node \( k, \) then the target is to compute a state prediction \( \hat{x}_{k,t} \) of \( x_t \) which is as similar as possible to the global state prediction.

Assume that the node \( k \) can collect the measurements from its neighbors \( \mathcal{N}_k. \) Then, the corresponding local nominal state-space model is

\[
\begin{align*}
x_{t+1} &= A x_t + B w_t + r_t \\
y_{l,t} &= C_l x_t + D_l v_{l,t}, \quad l \in \mathcal{N}_k.
\end{align*}
\]

(9)

The latter can be rewritten in the compact form

\[
\begin{align*}
x_{t+1} &= A x_t + \Gamma_B u_{kl}^{loc} + r_t \\
y_{loc, t} &= C_k^{loc} x_t + \Gamma_{Dloc} u_{k,t}^{loc}
\end{align*}
\]

(10)

where \( u_{k,t}^{loc} = [v_{l,t}^{T} (v_{l,t}^{loc})^{T}]^{T} \) is the input noise and \( y_{loc, t} \) is the output; \( v_{l,t}^{loc} \) and \( y_{loc, t} \) are given by stacking \( v_{l,t} \) and \( y_{l,t} \), with \( l \in \mathcal{N}_k, \) respectively. Moreover, \( C_k^{loc} \) is given by stacking \( C_l \) with \( l \in \mathcal{N}_k, \) \( \Gamma_{Dloc} = [0 D_k^{loc}] \) and \( D_k^{loc} \) is a block diagonal matrix whose main blocks are \( D_l \) with \( l \in \mathcal{N}_k, \) in addition, defining \( R_k^{loc} := D_k^{loc} (D_k^{loc})^T \) and \( S_k := (C_k^{loc})^T (R_k^{loc})^{-1} C_k^{loc} \) it results that

\[
S_k = \sum_{l \in \mathcal{N}_k} C_l^T R_l^{-1} C_l.
\]

We conclude that the one-step ahead predictor of \( x_t \) at node \( k \) is similar to the one in Equation (8) but now we need to discard the terms for which \( l \not\in \mathcal{N}_k. \) It is worth noting that the latter represents an intermediate local prediction of \( x_{t+1} \) at node \( k, \) and it is denoted as \( \psi_{k,t+1}. \) Allowing that the connected nodes can exchange their intermediate estimates, then each node can update the prediction at node \( k \) in terms of both \( \psi_{k,t+1} \) and \( \psi_{l,t+1} \) with \( l \in \mathcal{N}_k. \) More precisely, consider a matrix \( W = [w_{lk}]_{lk} \in \mathbb{R}^{N \times N} \) such that

\[
w_{lk} \geq 0 \text{ and } w_{lk} = 0 \text{ if } l \not\in \mathcal{N}_k \\
\sum_{l \in \mathcal{N}_k} w_{lk} = 1.
\]

(11)

Therefore, the final predicted state at node \( k \) is given by means of the so-called diffusion step [14]:

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

To sum up, in the diffusion scheme, each local unit uses the measurements and the intermediate local predictions from its neighbors. The resulting scheme is explained through Algorithm 1.
Algorithm 1 Distributed robust Kalman filter with uniform local tolerance at time $t$.

**Input:** $\hat{x}_{k,t}$, $V_{k,t}$, $y_{k,t}$, $W = [w_{ik}]_{ik}$ with $k = 1 \ldots N$

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

**Incremental step.** Compute at every node $k$:

$$\psi_{k,t+1} = Ax_{k,t} + A \left(V_{k,t}^{-1} + S_k\right)^{-1} \sum_{l \in N_k} C_l^T R_l^{-1} \left(y_{l,t} - C_l \hat{x}_{k,t}\right) + r_t$$

$$P_{k,t+1} = A \left(V_{k,t}^{-1} + S_k\right)^{-1} A^T + BB^T$$

Find $\theta_{k,t}$ s.t. $\gamma \left(P_{k,t+1}^{-1}, \theta_{k,t}\right) = c$

$$V_{k,t+1} = \left(P_{k,t+1}^{-1} - \theta_{k,t} I\right)^{-1}$$

**Diffusion step.** Compute at every node $k$:

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

It is worth noting that $\psi_{k,t}$ is computed by using the robust Kalman scheme in Equation (4) applied to the local model in Equation (10). In addition, $c$ is the same for any node that is $c$ takes a uniform value over the sensor network. In particular, the tolerance $c$ is the same for both the centralized and the distributed Kalman filter. This strategy for the selection of the tolerance does not ensure that the least favorable model computed at node $k$ is compatible with the one of the centralized filter. However, in the case of large deviations of the least favorable model corresponding to the centralized problem, it is very likely that the predictor at node $k$ using Algorithm 1 is better than the one which assumes that the nominal and actual models coincide. Finally, in the case that $c = 0$, i.e., the nominal model coincides with the actual one, Algorithm 1 boils down to the distributed Kalman filter with diffusion step in [14].

4. Distributed Robust Kalman Filtering with Non-Uniform Local Tolerance

We investigate the possibility to assign a possibly different local tolerance to each node that is the local tolerance is not uniform across the sensor network. Recall that the least favorable model is given by the minimax problem in Equation (3), with constant tolerance $c$, and the corresponding optimal estimator is given by the centralized robust Kalman filter in Equation (4).

Consider the centralized problem in Equation (3). Let

$$\tilde{f}_1 (z_t | Y_{t-1}) = \int \phi_t (z_t | x_t) \tilde{f}_1 (x_t | Y_{t-1}) \, dx_t$$

$$\tilde{f}_1 (z_t | Y_{t-1}) = \int \phi_t (z_t | x_t) \tilde{f}_1 (x_t | Y_{t-1}) \, dx_t$$

denote the pseudo-nominal and the least favorable conditional probability densities of $z_t$ given the past observations $Y_{t-1}$, respectively. Recall that $\phi_t (z_t | x_t)$ is the nominal transition density of the state space model in Equation (1) and thus

$$\phi_t (z_t | x_t) \sim N \left( \begin{bmatrix} A x_t + r_t \\ C x_t \end{bmatrix}, \begin{bmatrix} \Gamma_B & \Gamma_D \\ \Gamma_B^T & \Gamma_D^T \end{bmatrix} \right).$$

Since $\tilde{f}_1 (x_t | Y_{t-1}) \sim N (\bar{x}_t, V_t)$, and in view of Equations (14) and (16), we have

$$\tilde{f}_1 (z_t | Y_{t-1}) \sim N (m_z, K_z)$$

where
where
\[
m_{z_t} = \begin{bmatrix} A\hat{x}_t + r_t \\ C\hat{x}_t \end{bmatrix}, \quad K_{z_t} = \begin{bmatrix} A \\ C \end{bmatrix} V_t \begin{bmatrix} A^T & C^T \end{bmatrix} + \begin{bmatrix} \Gamma_B & \Gamma_D \\ \Gamma_D^T & \Gamma_D \end{bmatrix}.
\]  

(17)

In [29], it has been shown that the optimal solution \( \phi^0_t(z_t | x_t) \) to Equation (3) is Gaussian. Accordingly, in view of Equation (15), the corresponding least favorable density of \( z_t \) given \( Y_{t-1} \) is Gaussian:
\[
\hat{f}_t(z_t | Y_{t-1}) \sim \mathcal{N}(\hat{m}_{z_t}, \hat{K}_{z_t}).
\]

It is clear then that the minimax problem in Equation (3) can be written by replacing \( \phi_t(z_t | x_t) \) and \( \phi_t(z_t | x_t) \) with \( \hat{f}_t(z_t | Y_{t-1}) \) and \( \hat{f}_t(z_t | Y_{t-1}) \), respectively. Then, the equivalent minimax problem is
\[
\hat{x}_{t+1} = \arg\min_{\bar{g}_t \in \mathcal{G}_i} \max_{\hat{f}_t \in \mathcal{B}_t} \int \| x_{t+1} - g_t \|^2 \hat{f}_t(z_t | Y_{t-1}) dz_t
\]

(18)

where the ambiguity set is a ball about the pseudo-nominal density \( \hat{f}_t(z_t | Y_{t-1}) \)
\[
\mathcal{B}_t = \left\{ \hat{f}_t(z_t | Y_{t-1}) \sim \mathcal{N}(\bar{m}_{z_t}, \bar{K}_{z_t}) \text{ s.t. } D_{KL}(\hat{f}_t \| \bar{f}_t) \leq c \right\}
\]
formed by the KL divergence between \( \hat{f}_t(z_t | Y_{t-1}) \) and \( \hat{f}_t(z_t | Y_{t-1}) \):
\[
D_{KL}(\hat{f}_t \| \bar{f}_t) = \int \hat{f}_t(z_t | Y_{t-1}) \log \left( \frac{\hat{f}_t(z_t | Y_{t-1})}{\bar{f}_t(z_t | Y_{t-1})} \right) dz_t.
\]

Since \( \hat{f}_t(z_t | Y_{t-1}) \) and \( \hat{f}_t(z_t | Y_{t-1}) \) are Gaussian distributed, we have
\[
D_{KL}(\hat{f}_t \| \bar{f}_t) = \frac{1}{2} \left[ \| m_{z_t} - \bar{m}_{z_t} \|^2_{\bar{K}_{z_t}} - \log |\bar{K}_{z_t}| + \log |K_{z_t}| + \text{tr} \left( \bar{K}_{z_t} K_{z_t}^{-1} \right) - (n + p) \right].
\]

It is well known that \( D_{KL}(\hat{f}_t \| \bar{f}_t) \) also represents the negative log-likelihood of the model \( \hat{f}_t \) under the actual model \( \bar{f}_t \). Accordingly, \( c \) represents an upper bound of the negative log-likelihood and it can be found as follows. Fix the nominal state space model \( (A, B, C, D) \) and collect the data \( (y^N, u^N, x^N) \) where \( y^N = \{y_1 \ldots y_N\}, u^N = \{u_1 \ldots u_N\}, x^N = \{x_1 \ldots x_N\} \). Let \( \ell(A, B, C, D; y^N, u^N, x^N) \) be the negative log-likelihood of this nominal model using the collected data. Then, fix \( c = \ell(A, B, C, D; y^N, u^N, x^N) \). Clearly, we need to assume that the state is accessible to observation (or its estimate is reasonably good) to compute \( c \).

**Theorem 1** (Levy & Nikoukhah [30]). Let \( \hat{f}_t(z_t | Y_{t-1}) \) be the nominal density with mean \( m_{z_t} \) and covariance matrix \( K_{z_t} \) partitioned as
\[
K_{z_t} = \begin{bmatrix} K_{x_{t+1}} & K_{x_{t+1}, y_t} \\ K_{y_t, x_{t+1}} & K_{y_t} \end{bmatrix}
\]
according to the dimension of \( x_{t+1} \) and \( y_t \), respectively. The least favorable density \( \hat{f}^0_t(z_t | Y_{t-1}) \) solution to Equation (18) has mean and covariance matrix as follows:
\[
\hat{m}_{z_t} = m_{z_t}, \quad \hat{K}_{z_t} = \begin{bmatrix} K_{x_{t+1}} & K_{x_{t+1}, y_t} \\ K_{y_t, x_{t+1}} & K_{y_t} \end{bmatrix}.
\]

(19)
denote the nominal and least favorable error covariance matrices of \(x_{t+1}\) given \(Y_t\). Then,

\[
V_{t+1} = (P_{t+1}^{-1} - \theta_t I)^{-1}
\]  

(20)

and \(\theta_t > 0\) is the unique value for which

\[
D_{KL} \left( \tilde{f}_t^n || f_t \right) = \frac{1}{2} \left[ - \log |\hat{K}_{z_t}| + \log |K_{z_t}| + \text{tr} \left( \hat{K}_{z_t} K_{z_t}^{-1} \right) - (n + p) \right] = c. 
\]  

(21)

The above result provides a way to compute \(\tilde{f}_t^n(z_t|Y_{t-1})\). Indeed, once the centralized robust Kalman filter in Equation (4) has been computed, the mean and the covariance matrix of \(\tilde{f}_t^n(z_t|Y_{t-1})\) are given, in view of Equation (17), by

\[
\begin{align*}
\hat{m}_{z_t} &= \begin{bmatrix} A\hat{x}_t + r_t \\ C\tilde{x}_t \end{bmatrix}, \\
\hat{K}_{z_t} &= \begin{bmatrix} V_{t+1} + K_{x_{t+1};y_t} K_{y_t}^{-1} K_{y_t; x_{t+1}} & K_{y_t; x_{t+1}} \\
K_{y_t; x_{t+1}} & K_{y_t} \end{bmatrix}
\end{align*}
\]

(22)

where

\[
\begin{align*}
K_{x_{t+1};y_t} &= AV_t C^T \\
K_{y_t} &= CV_t C^T + \Gamma_D \Gamma_D^T.
\end{align*}
\]

From \(\tilde{f}_t(z_t|Y_{t-1})\) and \(\tilde{f}_t^n(z_t|Y_{t-1})\), we can compute the nominal and least favorable density for each node. Consider the state-space model in Equation (10) for node \(k\). Let \(z_{k,t} := [x_{t+1}^{\text{loc}} (y_{k,t}^{\text{loc}})^T]^T\). Then, the nominal transition probability at node \(k\), in view of Equation (10), is

\[
\phi_{k,t}(z_{k,t}|x_t) \sim \mathcal{N} \left( \begin{bmatrix} A\hat{x}_t + r_t \\ C\tilde{x}_t \end{bmatrix}, \begin{bmatrix} \Gamma_B & \Gamma_D \Gamma_D^T \\
\Gamma_D \Gamma_D^T & \Gamma_D \Gamma_D^T \end{bmatrix} \right).
\]  

(23)

Then,

\[
\tilde{f}_{k,t}(z_{k,t}|Y_{t-1}) = \int \phi_{k,t}(z_{k,t}|x_t) \tilde{f}_t(x_t|Y_{t-1}) \, dx_t
\]

denotes the pseudo-nominal conditional probability density of \(z_{k,t}\) given the past observations \(Y_{t-1}\) at node \(k\). Since \(\tilde{f}_t(x_t|Y_{t-1}) \sim \mathcal{N}(\hat{x}_t, V_t)\), and in view of Equation (23), we have

\[
\tilde{f}_{k,t}(z_{k,t}|Y_{t-1}) \sim \mathcal{N} \left( m_{z_{k,t}}, K_{z_{k,t}} \right)
\]

where

\[
\begin{align*}
m_{z_{k,t}} &= \begin{bmatrix} A\hat{x}_t + r_t \\ C\tilde{x}_t \end{bmatrix}, \\
K_{z_{k,t}} &= \begin{bmatrix} K_{x_{t+1}; y_{k,t}} & K_{x_{t+1}; y_{k,t}} \\
K_{y_{k,t}; x_{t+1}} & K_{y_{k,t}} \end{bmatrix}
\end{align*}
\]

(24)

and

\[
\begin{align*}
K_{x_{t+1}; y_{k,t}} &= AV_t (C_k^{\text{loc}})^T \\
K_{y_{k,t}; x_{t+1}} &= CV_t (C_k^{\text{loc}})^T + \Gamma_{D_k^{\text{loc}}} \Gamma_{D_k^{\text{loc}}}^T
\end{align*}
\]

(25)

Such a result is not surprising, indeed \(\tilde{f}_{k,t}(z_{k,t}|Y_{t-1})\) is given by marginalizing \(f_t(z_t|Y_{t-1})\) with respect to \(y_{l,t}\) with \(l \notin N_k\). Roughly speaking, this means that \(m_{z_{k,t}}, K_{x_{t+1}; y_{k,t}}\) and \(K_{y_{k,t}; x_{t+1}}\) are obtained from \(m_{z_t}, K_{x_{t+1}; y_t}\) and \(K_{y_t}\) as follows:

- \(m_{z_{k,t}}\) is the vector obtained from \(m_{z_t}\) by deleting the elements from \(pl - p + 1\) to \(pl\) for any \(l \notin N_k\).
• $K_{x_{t+1},y_{t}}$ is the matrix obtained from $K_{x_{t+1},y_{t}}$ by deleting the columns from $pl - p + 1$ to $pl$ for any $l \notin \mathcal{N}_k$.

• $K_{y_{t}}$ is the matrix obtained from $K_{y_{t}}$ by deleting the rows and the columns from $pl - p + 1$ to $pl$ for any $l \notin \mathcal{N}_k$.

Accordingly, we can compute the least favorable density at node $k$, say $\tilde{f}_{k,t}^{(0)}(z_{k,t} | Y_{t-1})$, by marginalizing $\tilde{f}_{k,t}^{(0)}(z_{t} | Y_{t-1})$ with respect to $y_{t,l}$ with $l \notin \mathcal{N}_k$. Therefore, we have

$$\tilde{f}_{k,t}^{(0)}(z_{k,t} | Y_{t-1}) \sim \mathcal{N} \left( m_{z_{k,t}}, \tilde{K}_{z_{k,t}} \right)$$

with

$$\tilde{K}_{z_{k,t}} = \begin{bmatrix} \tilde{K}_{x_{t+1}} & K_{x_{t+1},y_{t}} \\ K_{y_{t},x_{t+1}} & K_{y_{t}} \end{bmatrix} = \begin{bmatrix} V_{t+1} + K_{x_{t+1},y_{t}}K_{y_{t}}^{-1} & K_{x_{t+1},y_{t+1}} \\ K_{y_{t+1},x_{t+1}} & K_{y_{t+1}} \end{bmatrix}$$

where in the last equality we exploit Equation (22). It remains to design the robust filter to compute the intermediate prediction $\psi_{k,t+1}$.

**Remark 2.** At this point, it is worth doing a digression about Algorithm 1. The intermediate prediction at node $k$ is the solution to the following minimax problem

$$\psi_{k,t+1} = \arg\min_{g_{k,t} \in \mathcal{G}_{k,t}} \max_{f_{k,t} \in \mathbb{B}_{k,t}} \int \left\| x_{t+1} - g_{k,t}(y_{k,t}^{loc}) \right\|^2 \tilde{f}_{k,t}^{(0)}(z_{k,t} | Y_{t-1}) \, dz_{k,t}$$

where

$$\mathbb{B}_{k,t} := \{ f_{k,t} \ \text{s.t.} \ \mathcal{D}_{KL}(f_{k,t}^{*} || f_{k,t}) \leq c \}$$

and $\mathcal{G}_{k,t}$ is the set of all estimators $g_{k,t}$ whose variance is finite under any model in the ambiguity set $\mathbb{B}_{k,t}$. Moreover, in view of Theorem 1, the least favorable density $\tilde{f}_{k,t}^{(0)}(z_{k,t} | Y_{t-1})$ solution to Equation (26) is such that $\mathcal{D}_{KL}(\tilde{f}_{k,t}^{*} || \tilde{f}_{k,t}) = c$. It is worth noting that the best estimator at node $k$ would be the one constructed from $\tilde{f}_{k,t}^{(0)}$. On the other hand, the problem in Equation (26) implies neither $\tilde{f}_{k,t}^{(0)} = \tilde{f}_{k,t}^{*}$ nor $\mathcal{D}_{KL}(\tilde{f}_{k,t}^{*} || \tilde{f}_{k,t}) = \mathcal{D}_{KL}(\tilde{f}_{k,t}^{(0)} || \tilde{f}_{k,t})$.

Clearly, one would design the intermediate estimator at node $k$ by using $\tilde{f}_{k,t}^{(0)}$. However, the latter is not available at node $k$, and it is only known by a “central unit”, i.e., a unit knowing the global model, but neither collecting measurements nor computing predictions. Moreover, the transmission of the mean and the covariance matrix of $\tilde{f}_{k,t}^{(0)}$ would be more expensive in terms of transmission costs. As alternative, we can consider a minimax problem whose least favorable model $\tilde{f}_{k,t}^{(0)}$ is such that $\mathcal{D}_{KL}(\tilde{f}_{k,t}^{*} || f_{k,t}) = \mathcal{D}_{KL}(\tilde{f}_{k,t}^{(0)} || f_{k,t})$:

$$\psi_{k,t+1} = \arg\min_{g_{k,t} \in \mathcal{G}_{k,t}} \max_{f_{k,t} \in \mathbb{B}_{k,t}} \int \left\| x_{t+1} - g_{k,t}(y_{k,t}^{loc}) \right\|^2 \tilde{f}_{k,t}^{(0)}(z_{k,t} | Y_{t-1}) \, dz_{k,t}$$

where

$$\mathbb{B}_{k,t} := \{ f_{k,t} \ \text{s.t.} \ \mathcal{D}_{KL}(f_{k,t}^{*} || f_{k,t}) \leq c_{k,t} \}$$

$$c_{k,t} := \frac{1}{2} \left[ \log | \tilde{K}_{z_{k,t}} | + \log | K_{z_{k,t}} | + \text{tr} \left( \tilde{K}_{z_{k,t}} K_{z_{k,t}}^{-1} \right) - (n + p_k) \right]$$

and $p_k$ coincides with the number of rows of $Y_{k,t}^{loc}$. Under the above scheme, the central unit only transmits the local tolerance to each node in the network. The procedure which implements this optimized strategy of distributed robust Kalman filtering is outlined in Algorithm 2.
Algorithm 2 Distributed robust Kalman filter with non-uniform local tolerance at time $t$.

**Input:** $\hat{x}_{k,t}, V_{k,t}, y_{k,t}$, $W = [w_{j|k}]_{k}$ with $k = 1 \ldots N$

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

**Tolerance update.** Using the nominal global model, the central unit computes for every node $k$:

$$c_{k,t} = \frac{1}{2} \left[ -\log |K_{k,t}| + \log |K_{k,t+1}| + \text{tr} \left( K_{k,t} K_{k,t}^{-1} \right) - (n + p_{k}) \right] \quad (31)$$

**Incremental step.** Compute at every node $k$:

$$\psi_{k,t+1} = A \hat{x}_{k,t} + A \left( V_{k,t}^{-1} + S_{k} \right)^{-1} \sum_{l \in N_{k}} C_{l}^{T} R_{l}^{-1} (y_{l,t} - C_{l} \hat{x}_{l,t}) + r_{t}$$

$$p_{k,t+1} = A \left( V_{k,t}^{-1} + S_{k} \right)^{-1} A^{T} + BB^{T}$$

Find $\theta_{k,t}$ s.t. $\gamma (p_{k,t+1}, \theta_{k,t}) = c_{k,t}$

$$V_{k,t+1} = \left( p_{k,t+1}^{-1} - \theta_{k,t} I \right)^{-1}$$

**Diffusion step.** Compute at every node $k$:

$$\hat{x}_{k,t+1} = \sum_{l \in N_{k}} w_{lk} \psi_{l,t+1} \quad (33)$$

### 4.1. Least Favorable Performance

We show how to evaluate the performance of the previously introduced distributed algorithm with non-uniform local tolerance and diffusion step with respect to the least favorable model solution of the centralized problem in Equation (3). More precisely, we show how to compute the mean and the variance of the prediction error for each node $k$ in the network. In [29,34], it is shown that the least favorable model can be characterized through a state-space model over a finite interval $[0,T]$ as follows. Let $\xi_{t} = [x_{t}^{T} e_{t}^{T}]^{T}$, where $x_{t}$ is the least favorable state process. Then, the least favorable model takes the form

$$\dot{\xi}_{t+1} = \tilde{A}_{t} \xi_{t} + \tilde{B}_{t} \epsilon_{t} + \tilde{r}_{t}$$

$$y_{t} = \tilde{C}_{t} \xi_{t} + \tilde{D}_{t} \epsilon_{t} \quad (34)$$

where $\epsilon_{t}$ is normalized WGN, independent from $\hat{x}_{0}$, and $\tilde{r}_{t} := [r_{t}^{T} 0]^{T}$. Moreover,

$$\tilde{A}_{t} := \begin{bmatrix} A & \Gamma_{B} \Gamma_{H_{t}} \\ 0 & A - G_{t} C + (\Gamma_{B} - G_{t} \Gamma_{D}) \Gamma_{H_{t}} \end{bmatrix}, \quad \tilde{B}_{t} := \begin{bmatrix} \Gamma_{B} \Gamma_{L_{t}} \\ (\Gamma_{B} - G_{t} \Gamma_{D}) \Gamma_{L_{t}} \end{bmatrix}$$

$$\tilde{C}_{t} := \begin{bmatrix} C \\ \Gamma_{D} \Gamma_{H_{t}} \end{bmatrix}, \quad \tilde{D}_{t} := \Gamma_{D} \Gamma_{L_{t}}$$

where $\Gamma_{L_{t}}$ is such that $K_{t} = \Gamma_{L_{t}} \Gamma_{L_{t}}^{T}$

$$K_{t} := \left( I - (\Gamma_{B} - G_{t} \Gamma_{D})^{T} (\Omega_{t+1}^{-1} + \theta_{t} I) (\Gamma_{B} - G_{t} \Gamma_{D}) \right)^{-1}$$

$$\Gamma_{H_{t}} := K_{t} (\Gamma_{B} - G_{t} \Gamma_{D})^{T} (\Omega_{t+1}^{-1} + \theta_{t} I) (A - G_{t} C).$$

The matrix $\Omega_{t+1}^{-1}$ is computed from the backward recursion

$$\Omega_{t}^{-1} = (A - G_{t} C)^{T} (\Omega_{t+1}^{-1} + \theta_{t} I) (A - G_{t} C) + \Gamma_{L_{t}}^{T} K_{t}^{-1} \Gamma_{H_{t}}$$

where the final point is initialized with $\Omega_{T+1}^{-1} = 0$. 

Let \( \tilde{x}_{k,t} = x_t - \hat{x}_{t,k} \) denote the least favorable state prediction error \( \tilde{x}_{k,t} \) of node \( k \) at time \( t \) using Algorithm 2 or Algorithm 1. Define the vector containing all the errors across the network

\[
\mathbf{\tilde{x}}_t := \begin{bmatrix} \mathbf{\tilde{x}}_{1,t}^T & \cdots & \mathbf{\tilde{x}}_{N,t}^T \end{bmatrix}^T.
\]

Using the same reasonings in [39], it is not difficult to prove that \( \mathbf{\tilde{x}}_t \) obeys the following dynamics

\[
\mathbf{\tilde{x}}_{t+1} = \mathbf{A}_t \mathbf{\tilde{x}}_t + \mathbf{B}_t \mathbf{e}_t + \mathbf{C}_t \mathbf{e}_t
\]  
(35)

where

\[
\mathbf{A}_t := (W^T \otimes I)(I \otimes A)(\mathbf{V}_t^{-1} + \mathbf{S})^{-1}V_t^{-1}
\]
\[
\mathbf{B}_t := -(W^T \otimes I)(I \otimes A)(\mathbf{V}_t^{-1} + \mathbf{S})^{-1}(f^T \otimes I)\mathbf{C}^T \mathbf{R}^{-1} \mathbf{D} \mathbf{L}_t + 1 \otimes B \mathbf{N}_t
\]
\[
\mathbf{C}_t := -(W^T \otimes I)(I \otimes A)(\mathbf{V}_t^{-1} + \mathbf{S})^{-1}(f^T \otimes I)\mathbf{C}^T \mathbf{R}^{-1} \mathbf{D} \mathbf{H}_t + 1 \otimes B \mathbf{M}_t
\]
\[
\mathbf{\Gamma}_t := \text{diag}(\mathbf{C}_1, \ldots, \mathbf{C}_N)
\]
\[
\mathbf{V}_t := \text{diag}(\mathbf{V}_{1,t}, \ldots, \mathbf{V}_{N,t})
\]
\[
\mathbf{S} := \text{diag}(\mathbf{S}_1, \ldots, \mathbf{S}_N),
\]
\[
\mathbf{M}_t \in \mathbb{R}^{n \times n}, \mathbf{H}_t \in \mathbb{R}^{p \times n}, \mathbf{N}_t \in \mathbb{R}^{n \times (p \times n)} \text{ and } \mathbf{L}_t \in \mathbb{R}^{p \times (p \times n)} \text{ are such that } \mathbf{\Gamma}_t = [\mathbf{M}_t^T \mathbf{H}_t^T]^T \text{ and } \mathbf{\Gamma}_t = [\mathbf{N}_t^T \mathbf{L}_t^T]^T.
\]

Finally, \( \mathbf{1} \) denotes the vector of ones. Then, we combine Equation (35) with the model for \( \mathbf{e}_t \) in Equation (34):

\[
\mathbf{\eta}_{t+1} = \mathcal{F}_t \mathbf{\eta}_t + \mathcal{G}_t \mathbf{e}_t
\]  
(36)

where \( \mathbf{\eta}_t := [\mathbf{\tilde{x}}_t^T \mathbf{e}_t^T]^T \),

\[
\mathcal{F}_t := \begin{bmatrix} \mathbf{A}_t & \mathbf{C}_t \\
0 & (A - \mathbf{G}_t \mathbf{C}) + (\mathbf{G}_t \mathbf{D} - \mathbf{G}_t \mathbf{D}) \mathbf{\Gamma}_{H_t} \end{bmatrix}
\]
\[
\mathcal{G}_t := \begin{bmatrix} \mathbf{B}_t \\
(\mathbf{G}_t - \mathbf{G}_t \mathbf{D}) \mathbf{\Gamma}_{L_t} \end{bmatrix}
\]  
(37)

Taking the expectation of Equation (36), we obtain

\[
\mathbb{E}[\mathbf{\eta}_{t+1}] = \mathcal{F}_t \mathbb{E}[\mathbf{\eta}_t].
\]  
(38)

In view of the fact that \( \mathbf{x}_0 \) has mean equal to \( \hat{x}_0 \) and \( \tilde{x}_{k,0} = \hat{x}_0 \) for \( k = 1 \ldots N \), it is not difficult to see that \( \mathbb{E}[\mathbf{\eta}_0] = 0 \). This implies that \( \mathbf{\eta}_t \) is a zero mean stochastic process or, equivalently, all the predictors are unbiased. Next, we show how to derive the variance of the prediction errors. Let \( \mathcal{Q}_t = \mathbb{E}[\mathbf{\eta}_t \mathbf{\eta}_t^T] \).

In view of the fact that \( \mathbf{e}_t \) is normalized WGN, by Equation (36), we have that \( \mathcal{Q}_t \) is given by solving the following Lyapunov equation

\[
\mathcal{Q}_{t+1} = \mathcal{F}_t \mathcal{Q}_t \mathcal{F}_t^T + \mathcal{G}_t \mathcal{G}_t^T
\]  
(39)

We partition \( \mathcal{Q}_t \) as follows:

\[
\mathcal{Q}_t = \begin{bmatrix} \mathcal{P}_t & \mathcal{H}_t \\
\mathcal{H}_t^T & \mathcal{R}_t \end{bmatrix}
\]  
(40)
where $P_t \in \mathbb{R}^{N_n \times N_n}$, $H_t \in \mathbb{R}^{N_n \times n}$ and $R_t \in \mathbb{R}^{n \times n}$. Notice that $P_t$ contains in the main block diagonal the covariance matrices of the estimation error at each node. Accordingly, the least favorable mean square deviation is given by

\[
\text{MSD}_t := \frac{1}{N} \sum_{k=1}^{N} \text{MSD}_{k,t} = \frac{1}{N} \text{tr}(P_t)
\]

where MSD$_{k,t}$ is the variance of the prediction error at node $k$. Finally, we have the following convergence result for the proposed distributed algorithm.

**Proposition 1.** Let $(A, B)$ be a reachable pair and $(A, C_k^{loc})$ be an observable pair for any $k$. Let $W$ be an arbitrary diffusion matrix satisfying Equation (11). Then, there exists $c > 0$ sufficiently small such that, for any arbitrary initial condition $V_0 > 0$ and $V_{k,0} > 0$, the sequence $Q_t$, $t \geq 0$, generated by Equation (39) converges to $\bar{Q} > 0$ over $[\alpha T, \beta T]$ as $T \to \infty$. Moreover, we have $F_l \rightarrow F$, $G_l \rightarrow G$, and $c_{k,l} \rightarrow c_k$. In particular, $\bar{Q}$ corresponds to the unique solution of the algebraic Lyapunov equation

\[
\bar{Q} = F \bar{Q} F^T + G G^T
\]

with $F$ Schur stable. Accordingly, $\overline{\text{MSD}}_t$ converges over $[\alpha T, \beta T]$ as $T \to \infty$.

**Proof.** First, notice that the observability condition on the pairs $(A, C_k^{loc})$ implies the observability of $(A, C)$. Since the global model is reachable and observable, the robust centralized Kalman filter converges provided that $c$ is sufficiently small (see [43,44]). As a consequence, $V_l \rightarrow \bar{V} > 0$ as $t \to \infty$. Accordingly, in view of Equation (17), $K_{z_l} \rightarrow \bar{K}_z$ and, thus, in view of Equation (22), $\bar{K}_{z_{l_j}} \rightarrow \bar{K}_z$. Since $K_{z_{l_j}}$ and $\bar{K}_{z_{l_j}}$ are submatrices of $K_{z_l}$ and $\bar{K}_{z_l}$, respectively, we have that $K_{z_{l_j}} \rightarrow K_{z_l}$ and $\bar{K}_{z_{l_j}} \rightarrow \bar{K}_z$. Accordingly, in view of Equation (30), we have that $c_{k,l} \rightarrow \bar{c}_k$ where

\[
\bar{c}_k := \frac{1}{2} \left[- \log |\bar{K}_{z_l}| + \log |K_{z_l}| + \text{tr} \left( \bar{K}_{z_l} K_{z_l}^{-1} \right) - (n + p_k) \right].
\]

In [30], it has been shown that $V_l \rightarrow P_l$ as $c \to 0$, and thus $\bar{K}_{z_l} \rightarrow \bar{K}_z$. Since $K_{z_{l_j}}$ and $\bar{K}_{z_{l_j}}$ are submatrices of $K_{z_l}$ and $\bar{K}_{z_l}$, respectively, we have that $\bar{K}_{z_{l_j}} \rightarrow K_{z_{l_j}}$. Accordingly, in view of Equation (30), we have that $c_{k,l} \rightarrow 0$ as $c \to 0$.

In view of [43], Proposition 5.3, we conclude that the robust local Kalman filter at node $k$ converges because: the local state-space model is reachable and observable; $\bar{c}_k$ is sufficiently small provided that $c$ is sufficiently small as well.

Finally, the remaining part of the proof follows the one in ([39], Section IV-A) (see also [45]).

**5. Numerical Examples**

In this section, we test the performance of the distributed Kalman filters with uniform versus non-uniform local tolerance. More precisely, we consider the problem in [39] to track the position of a projectile from position observations corrupted by noise and coming from a network of $N = 20$ sensors. The latter is shown in Figure 1.
The model for the projectile motion is

\[ \dot{x}_t^c = \Phi x_t^c + r_t^c \]  

(43)

where

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

\[ r_t^c = \begin{bmatrix} 0 & 0 & -g & 0 & 0 \end{bmatrix}^T, \text{ with } g = 10, \]

and \( x_t^c = \begin{bmatrix} v_{x,t} & v_{y,t} & v_{z,t} & p_{x,t} & p_{y,t} & p_{z,t} \end{bmatrix}^T \), with \( v \) denoting the velocity and \( p \) the position in the three spatial components. We discretize Equation (43) by using a sampling time equal to 0.1 s. In this way, the model becomes

\[ x_{t+1} = Ax_t + r_t \]

where \( x_t \) is the sampled version of \( x_t^c \), \( A = I_6 + 0.1\Phi \) and \( r_t = (0.1I_6 + 0.1^2\Phi/2)r_t^c \). Assuming that every sensor can measure only along two spatial components, the output matrix of the \( k \)th node can be of the type

\[ C_k = \begin{bmatrix} 0 & 0 & 0 & \text{diag}(1, 1, 0) \end{bmatrix}, \quad C_k = \begin{bmatrix} 0 & 0 & 0 & \text{diag}(1, 0, 1) \end{bmatrix}, \quad C_k = \begin{bmatrix} 0 & 0 & 0 & \text{diag}(0, 1, 1) \end{bmatrix}. \]

Every output matrix is assigned randomly to each node, with the constraint that the local state-space model in Equation (10) associated to each node must be observable for every node to guarantee the convergence of the robust Kalman filters at every node. Thus, if any node violates the constraint, all the output matrices are discarded and reassigned. Then, we choose \( B = \sqrt{0.001I} \), \( R_k = D_kD_k^T = \sqrt{kPR_0P^T} \), where \( R_0 = 0.5 \cdot \text{diag}(1, 4, 7) \) and \( P \) is a permutation matrix which is randomly generated for any node. The initial state \( x_0 \) is Gaussian distributed and whose covariance matrix is \( P_0 = I \).

In the numerical simulations, the following Kalman filters are considered: the centralized Kalman filter (KFC); the centralized robust Kalman filter (RKFC); the distributed Kalman filter with diffusion step (KFD) proposed in [14]; the distributed robust Kalman with diffusion step and uniform local tolerance (RKFDU) proposed in [39] and reviewed in Section 3; and the distributed robust Kalman filter with diffusion step and non-uniform local tolerance (RKFDNU) proposed in Section 4. For the distributed filters, the diffusion matrix \( W \) is defined as

\[ w_{lk} = \begin{cases} \alpha_k n_l, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \]
where \( n_l \) represents the total number of neighbors of node \( l \) while \( \alpha_k \) is chosen such that Equation (11) holds.

5.1. First Example

We assume that the actual model is contained in the ambiguity set Equation (2) with \( c = 0.02 \). Figure 2 shows the least favorable mean squared deviation across the network. We notice that \( \text{MSD}_t \) converges at the steady-state for all the distributed versions of the Kalman filter. RKFDNU performs slightly better that RKFDU and both perform consistently better than KFD. Finally, all of them perform worse than the centralized versions, and RKFC results the best.

![Figure 2](image-url)

**Figure 2.** Least favorable mean square deviation across the network with tolerance \( c = 0.02 \).

However, the situation is more salient if we consider the steady-state least favorable \( \text{MSD}_{k,t} \) for each node (see Figure 3a): RKFDNU performs slightly better than RKFDU for the majority of the nodes. However, there is a clear difference for nodes 18 and 19 which are more susceptible to model uncertainty: RKFDNU performs better than RKFDU.

![Figure 3](image-url)

**Figure 3.** (a) Least favorable mean square deviation for each node in the steady-state with tolerance \( c = 0.02 \). (b) Time-variant local tolerances for each node over time with \( c = 0.02 \).
Figure 3b shows the behavior of the local tolerances $c_{k,t}$ over the time for RKFDNU. As expected, every $c_{k,t}$ converges to a constant value. However, the latter is different from the tolerance $c$ of the centralized minimax problem.

Finally, Figure 4a,b shows the risk sensitivity parameters $\theta_{k,t}$ at every node for RKFDU and RKFDNU. We can observe that the risk sensitivity parameters of RKFDU takes larger values than the ones of RKFDNU. Accordingly, the inferior performance of RKFDU is due by the fact that the robust local filters are too conservative.

![Figure 4a](image1.png) ![Figure 4b](image2.png)

**Figure 4.** (a) Risk sensitivity parameter for each node using RKFDU with tolerance $c = 0.02$. (b) Risk sensitivity parameter for each node using RKFDNU with tolerance $c = 0.02$.

### 5.2. Second Example

In the second experiment, we consider a larger deviation between the actual model and the nominal one, i.e., we choose $c = 0.06$.

Figures 5 and 6a show the least favorable mean square deviation across the network and for each node in the steady-state. The situation is similar to the previous one, but the difference among RKFD, RKFDU, and RKFDNU is more evident. In particular, the steady-state value of $\overline{\text{MSD}}_{k,t}$ for $k = 18, 19$ using RKFDNU is clearly better than the ones corresponding to KFD and RKFDU.

In addition, Figure 6b shows the tolerances $c_{k,t}$ at every node over the time. As expected, the latter are higher than the ones with $c = 0.02$. Indeed, the uncertainty now is greater than before and thus the robust local filters now must be more conservative than before.

Finally, we study how the least favorable MSD for each node correlates with the topology of the sensor network. Figure 7a,b shows two additional sensor networks obtained from the original network of Figure 1 by adding connections only to some nodes. More precisely, the density of the original network, i.e., number of connections over all possible connections, is $d_1 = 0.39$; the density of the networks in Figure 7a,b is $d_2 = 0.48$ and $d_3 = 0.72$, respectively.
Figure 5. Least favorable mean square deviation across the network with tolerance $c = 0.06$.

Figure 6. (a) Least favorable mean square deviation for each node in the steady-state with tolerance $c = 0.06$. (b) Time-variant local tolerances for each node over time with $c = 0.06$.

Figure 7. (a) Sensor network with density $d_2 = 0.48$. (b) Sensor network with density $d_3 = 0.72$. 
Figure 8a,b shows the results obtained by RKFDNU with the three different sensor networks. As expected, the increase of the degrees of the nodes, and consequently of the connections in the network, reduces the least favorable MSD related to those nodes at steady-state and the total least favorable MSD across the network. In conclusion, by adding edges the performance of RKFDNU tends to the one obtained in the centralized case (RKFC), where the nodes are considered all connected to each other.

![Figure 8a](image1)

*Figure 8a. Performance of RKFDNU across the three networks compared with RKFC. (a) Least favorable mean square deviation with tolerance $c = 0.06$. (b) Least favorable mean square deviation for each node in the steady-state with tolerance $c = 0.06$."

6. Efficient Algorithm

Proposition 1 suggests a simplified version of Algorithm 2. Indeed, if $c$ is sufficiently small, then $c_{k,t}$ converges to $\bar{c}_k$ in the steady state for every node of the network. Accordingly, the central unit can compute $\bar{c}_k$ and transmit it to any node once. In this way, the transmission costs are reduced. The resulting procedure is outlined in Algorithm 3.

**Algorithm 3** Efficient distributed robust Kalman filter with non-uniform local tolerance at time $t$.

*Input:* $\hat{x}_{k,t}, V_{k,t}, y_{k,t}, W = [w_{lk}]_{lk}$ with $k = 1 \ldots N$

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

**Incremental step.** Compute at every node $k$:

$$\psi_{k,t+1} = A\hat{x}_{k,t} + A (V_{k,t}^{-1} + S_k)^{-1} \sum_{l \in N_k} C_l^T R_l^{-1} (y_{l,t} - C_l \hat{x}_{l,t}) + r_t$$

$$P_{k,t+1} = A (V_{k,t}^{-1} + S_k)^{-1} A^T + BB^T$$

Find $\theta_{k,t}$ s.t. $\gamma (P_{k,t+1}, \theta_{k,t}) = \bar{c}_k$

$$V_{k,t+1} = (P_{k,t+1}^{-1} - \theta_{k,t} I)^{-1}$$

**(44)**

**Diffusion step.** Compute at every node $k$:

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

We compared this algorithm, hereafter called RKFDNU2, with RKFDNU: the performance in practice is the same. Figure 9 shows their least favorable mean square deviation across the network.
in the scenario of Section 5.2 in the first 50 time steps. Finally, Figure 10a,b shows the risk sensitivity parameters for RKDFNU and RKFDNU2, respectively: there is a slight difference. However, we saw that such a difference disappears after 20 time steps. We conclude that the efficient scheme RKFDNU2 represents a good approximation of RKFDNU.

Figure 9. Least favorable mean square deviation across the network with tolerance $c = 0.02$.

Figure 10. (a) Risk sensitivity parameter for each node using RKFDNU with tolerance $c = 0.02$. (b) Risk sensitivity parameter for each node using RKFDNU2 with tolerance $c = 0.02$.

Finally, Table 1 summarizes the performance of RKFC, RKFDU, RKFDNU, and RKFDNU2 obtained with tolerance $c = 0.02$. The considered values are the least favorable MSD across the network at steady-state, the average among every node of the tolerances at steady-state, the average among every node of the risk sensitivity parameter at steady-state, and the occurred communications
between the central unit and the local nodes in the whole time span. In particular, concerning the communication:

- in RKFDU, the central unit transmits the uniform tolerance to each node at once (at the beginning);
- in RKFDNU, the central unit transmits the local tolerances to each node at every time step;
- in RKFDNU2, the central unit transmits the steady-state local tolerances to each node at once (at the beginning).

Table 1. Summary of the performance of the different algorithms with tolerance $c = 0.02$.

|                | RKFC | RKFDU | RKFDNU | RKFDNU2 |
|----------------|------|-------|--------|---------|
| MSD [dB]       | −2.9174 | −1.5553 | −1.6182 | −1.6182 |
| Average tolerance | 0.02  | 0.02  | 0.0144 | 0.0144 |
| Average risk sensitivity parameter | 1.3366 | 0.3764 | 0.3576 | 0.3576 |
| Communication requirement | N.A. | One at the beginning | Every time step | One at the beginning |

7. Conclusions

In this article, the problem of distributed robust Kalman filtering for a sensor network is considered. More precisely, we consider a distributed scheme with diffusion step and the intermediate estimate is designed in order to be optimal according to the least favorable model belonging to a prescribed local ambiguity set. The latter is a ball about the local nominal model and the radius of this ball is the local tolerance. In this paper, we propose an algorithm in which the local tolerance of each node is different and suitably computed by the central unit. We also consider a more efficient implementation of the algorithm where the central unit computes and transmits the steady-state local tolerances for every node at once. In this way, the communication between the central unit and the local nodes is reduced. Through some numerical examples, we showed that the proposed algorithm performs better than the one with a uniform local tolerance across the network.

Author Contributions: Conceptualization, A.E., F.G., G.G. and M.Z.; methodology, A.E., F.G., G.G. and M.Z.; simulations, A.E., F.G., G.G. and M.Z.; supervision, M.Z.; All the authors contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: Part of this work has been supported by the MIUR (Italian Minister for Education) under the initiative “Departments of Excellence” (Law 232/2016).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, H.N.; Wang, H.D. Distributed consensus observers-based $H_{\infty}$ control of dissipative PDE systems using sensor networks. *IEEE Trans. Control Netw. Syst.* 2015, 2, 112–121. [CrossRef]
2. Orihuela, L.; Millán, P.; Vivas, C.; Rubio, F.R. Distributed control and estimation scheme with applications to process control. *IEEE Trans. Control Syst. Technol.* 2015, 23, 1563–1570. [CrossRef]
3. Sadamoto, T.; Ishizaki, T.; Imura, J. Average state observers for large-scale network systems. *IEEE Trans. Control Netw. Syst.* 2017, 4, 761–769. [CrossRef]
4. Wang, S.; Ren, W.; Chen, J. Fully distributed dynamic state estimation with uncertain process models. *IEEE Trans. Control Netw. Syst.* 2017, 5, 1841–1851. [CrossRef]
5. Dormann, K.; Noack, B.; Hanebeck, U. Optimally distributed Kalman filtering with data-driven communication. *Sensors* 2018, 18, 1034. [CrossRef] [PubMed]
6. Ruan, Y.; Luo, Y.; Zhu, Y. Globally optimal distributed Kalman filtering for multisensor systems with unknown inputs. *Sensors* 2018, 18, 2976. [CrossRef]
7. Gao, S.; Chen, P.; Huang, D.; Niu, Q. Stability analysis of multi-sensor Kalman filtering over lossy networks. *Sensors* 2016, 16, 566. [CrossRef]
8. Kordestani, M.; Samadi, M.; Saif, M. A distributed fault detection and isolation method for multifunctional spoiler system. In Proceedings of the 2018 IEEE 61st International Midwest Symposium on Circuits and Systems, Windsor, ON, Canada, 5–8 August 2018; pp. 380–383.

9. Scott, S.; Blocker, A.; Bonassi, F.; Chipman, H.; George, E.; McCulloch, R. Bayes and big data: The consensus Monte Carlo algorithm. *EFAbBayes 250th Conf.* 2013, 16. [CrossRef]

10. Spanos, D.P.; Olfati-Saber, R.; Murray, R.M. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, Boise, ID, USA, 15 April 2005; pp. 133–139.

11. Olfati-Saber, R. Distributed Kalman filter with embedded consensus filters. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 8179–8184.

12. Olfati-Saber, R. Distributed Kalman filtering for sensor networks. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5492–5498.

13. Carli, R.; Chiuso, A.; Schenato, L.; Zampieri, S. Distributed Kalman filtering based on consensus strategies. *IEEE J. Sel. Areas Commun.* 2008, 26, 622–633. [CrossRef]

14. Cattivelli, F.S.; Sayed, A.H. Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Trans. Autom. Control* 2010, 55, 2069–2084. [CrossRef]

15. Yang, S.; Huang, T.; Guan, J.; Xiong, Y.; Wang, M. Diffusion Strategies for Distributed Kalman Filter with Dynamic Topologies in Virtualized Sensor Networks. *Mob. Inf. Syst.* 2016, 2016, 8695102. [CrossRef]

16. Ji, H.; Lewis, F.L.; Hou, Z.; Mikulski, D. Distributed information weighted Kalman consensus filter for sensor networks. *Automatica* 2017, 77, 18–30. [CrossRef]

17. Song, E.; Zhu, Y.; Zhou, J.; You, Z. Optimal Kalman filtering fusion with cross-correlated sensor noises. *Automatica* 2007, 43, 1450–1456. [CrossRef]

18. Xu, J.; Song, E.; Luo, Y.; Zhu, Y. Optimal distributed Kalman filtering fusion algorithm without invertibility of estimation error and sensor noise covariances. *IEEE Signal Process. Lett.* 2012, 19, 55–58. [CrossRef]

19. Sun, S.L.; Deng, Z.L. Multi-sensor optimal information fusion Kalman filter. *Automatica* 2004, 40, 1017–1023. [CrossRef]

20. Feng, J.; Zeng, M. Optimal distributed Kalman filtering fusion for a linear dynamic system with cross-correlated noises. *Int. J. Syst. Sci.* 2012, 43, 385–398. [CrossRef]

21. Andrieu, C.; Doucet, A.; Holenstein, R. Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B* 2010, 72, 269–342. [CrossRef]

22. Martino, L.; Elvira, V.; Camps-Valls, G. Distributed particle Metropolis-Hastings schemes. In Proceedings of the IEEE Statistical Signal Processing Workshop (SSP), Freiburg, Germany, 10–13 June 2018.
34. Zorzi, M. Robust Kalman filtering under model perturbations. *IEEE Trans. Autom. Control* **2017**, *62*, 2902–2907. [CrossRef]

35. Abadeh, S.; Nguyen, V.; Kuhn, D.; Esfahani, P. Wasserstein distributionally robust Kalman filtering. In *Advances in Neural Information Processing Systems*; MIT Press: Montreal, CA, USA, 2018; pp. 8474–8483.

36. Shen, B.; Wang, Z.; Hung, Y. Distributed $H_{\infty}$-consensus filtering in sensor networks with multiple missing measurements: The finite-horizon case. *Automatica* **2010**, *46*, 1682–1688. [CrossRef]

37. Luo, Y.; Zhu, Y.; Luo, D.; Zhou, J.; Song, E.; Wang, D. Globally optimal multisensor distributed random parameter matrices Kalman filtering fusion with applications. *Sensors* **2008**, *8*, 8086–8103. [CrossRef]

38. Huang, J.; Shi, D.; Chen, T. Distributed robust state estimation for sensor networks: A risk-sensitive approach. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 6378–6383.

39. Zorzi, M. Distributed Kalman filtering under model uncertainty. *IEEE Trans. Control Netw. Syst.* **2019**. [CrossRef]

40. Cover, T.; Thomas, J. *Information Theory*; Wiley: New York, NY, USA, 1991.

41. Zorzi, M. An interpretation of the dual problem of the THREE-like approaches. *Automatica* **2015**, *62*, 87–92. [CrossRef]

42. Zorzi, M. A new family of high-resolution multivariate spectral estimators. *IEEE Trans. Autom. Control* **2014**, *59*, 892–904. [CrossRef]

43. Zorzi, M.; Levy, B.C. On the convergence of a risk sensitive like filter. In Proceedings of the 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 4990–4995.

44. Zorzi, M. Convergence analysis of a family of robust Kalman filters based on the contraction principle. *SIAM J. Control Optim.* **2017**, *55*, 3116–3131. [CrossRef]

45. Zorzi, M.; Levy, B. Robust Kalman filtering: Asymptotic analysis of the least favorable model. In Proceedings of the 57th IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).