Distributing the Kalman Filter for Large-Scale Systems

Usman A. Khan, *Student Member, IEEE*, and José M. F. Moura, *Fellow, IEEE*

***Abstract*—This paper presents a *distributed* Kalman filter to estimate the state of a sparsely connected, large-scale, n-dimensional, dynamical system monitored by a network of N sensors. Local Kalman filters are implemented on n_l-dimensional subsystems, $n_l \ll n$, obtained by spatially decomposing the large-scale system. The distributed Kalman filter is optimal under an Lth order Gauss–Markov approximation to the centralized filter. We quantify the information loss due to this Lth-order approximation by the divergence, which decreases as L increases. The order of the approximation L leads to a bound on the dimension of the subsystems, hence, providing a criterion for subsystem selection. The (approximated) centralized Riccati and Lyapunov equations are computed iteratively with only local communication and low-order computation by a distributed iterate collapse inversion (DICI) algorithm. We fuse the observations that are common among the local Kalman filters using bipartite fusion graphs and consensus averaging algorithms. The proposed algorithm achieves full distribution of the Kalman filter. Nowhere in the network, storage, communication, or computation of n-dimensional vectors and matrices is required; only $n_l \ll n$ dimensional vectors and matrices are communicated or used in the local computations at the sensors. In other words, knowledge of the state is itself distributed.**

***Index Terms*—Distributed algorithms, distributed estimation, information filters, iterative methods, Kalman filtering, large-scale systems, matrix inversion, sparse matrices.**

I. Introduction

CENTRALIZED implementation of the Kalman filter [1], [2], although possibly optimal, is neither robust nor scalable to complex large-scale dynamical systems with their measurements distributed on a large geographical region. The reasons are twofold: i) the large-scale systems are very high-dimensional, and thus require extensive computations to implement the centralized procedure; and ii) the span of the geographical region, over which the large-scale system is deployed or the physical phenomenon is observed, poses a large communication burden and thus, among other problems, adds latency to the estimation mechanism. To remove the difficulties posed by centralization, we decompose the large-scale system into n_l-dimensional subsystems and distribute the estimation algorithm with a low order Kalman filter implemented at each of these subsystems. To account for the processing, communication, and limited resources at the subsystems, the local Kalman filters involve computations and communications with local quantities only, i.e., vectors and matrices of low dimensions, $n_l \ll n$, where n is the dimension of the state vector—no sensor computes, communicates, or stores any n-dimensional quantity.

Much of the existing research on distributed Kalman filters focuses on sensor networks monitoring *low* dimension systems. This research replicates an nth-order Kalman filter at each sensor, which is only practical, when the dimension of the state is small, for example, when multiple sensors mounted on a small number of robot platforms are used for target tracking [3]–[5]. The problem in such scenarios reduces to how to efficiently incorporate the distributed observations, which is also referred to in the literature as "data fusion"; see also [6]. Data fusion for Kalman filters over arbitrary communication networks is discussed in [7], using iterative consensus protocols provided in [8]. The consensus protocols in [8] are assumed to converge asymptotically; thus, between any two time steps of the Kalman filter, the consensus protocols require an infinite number of iterations to achieve convergence. It is worth mentioning that, with a finite number of iterations (true for any practical implementation), the resulting Kalman filter does not remain optimal. References [4] and [9] incorporate packet losses, intermittent observations, and communication delays in the data fusion process. Because they replicate an n-dimensional Kalman filter at each sensor, they communicate and invert $n \times n$ matrices locally, which, in general, is an $O(n^3)$ computation. This may be viable for low-dimensional systems, as in tracking, but unacceptable in the problems we consider where the state dimension n is very large, for example, in the range of 10^2 to 10^9. In such problems, replication of the global dynamics in the local Kalman filters is either not practical or not possible.

Kalman filters with reduced-order models have been studied in, e.g., [10] and [11] to address the computation burden posed by implementing nth-order models. In these works, the reduced models are decoupled, which is suboptimal, as important coupling among the system variables is ignored. Furthermore, the network topology is either fully connected [10] or is close to fully connected [11], requiring long-distance communication that is expensive. We are motivated by problems where the large-scale systems, although sparse, cannot be decoupled and where, due to the sensor constraints, the communication and computation should both be local.

Manuscript received July 31, 2007; revised April 22, 2008. First published June 20, 2008; current version published September 17, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aleksandar Dogandzic. This work was partially supported by the DARPA DSO Advanced Computing and Mathematics Program Integrated Sensing and Processing (ISP) Initiative under ARO Grant DAAD 19-02-1-0180, by the NSF under Grants ECS-0225449 and CNS-0428404, and by an IBM Faculty Award.

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: ukhan@ece.cmu.edu; moura@ece.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2008.927480

We present a distributed Kalman filter that addresses *both* the computation and communication challenges posed by complex large-scale dynamical systems, while preserving its coupled structure; in particular, nowhere in our distributed Kalman filter do we store, communicate, or compute any n-dimensional quantity. As an interesting remark, nowhere either in the network is there a copy of the entire state estimate; in other words, knowledge about the state is intrinsically distributed. We briefly explain the key steps and approximations in our solution.

1) Spatial Decomposition of Complex Large-Scale Systems: To distribute the Kalman filter, we provide a spatial decomposition of the complex large-scale dynamical system (of dimension n) that we refer to as *the overall system* into several, possibly many, local coupled dynamical systems (of dimension n_l, such that $n_l \ll n$) that we refer to as *subsystems* in the following. The large-scale systems we consider are sparse and localized. Physical systems with such characteristics are described in Section II-A, as resulting, for example, from a spatio–temporal discretization of random fields. These subsystems overlap, i.e., they share states, and thus the resulting local Kalman filters also overlap. In addition to this overlap, the subsystems are connected by local interactions that account for the coupling between the subsystems. We preserve this coupling by modeling explicitly this information exchange among the subsystems.

2) Overlapping Dynamics at the Subsystems: Bipartite Fusion Graphs: The subsystems that we extract from the overall system overlap. In particular, some state variables are observed by several subsystems. To fuse this shared information, we implement a fusion algorithm using bipartite fusion graphs, which we introduced in [12], and local average consensus algorithms [13]. The interactions required by the fusion procedure are constrained to a small neighborhood and with a particular choice of the communication topology the observation fusion procedure remains single hop.

3) Assimilation of the Local Error Covariances—Distributed Iterate-Collapse Inversion Algorithm: A key issue when distributing the Kalman filter is that the local error covariances approximate the centralized error covariances in a meaningful way. If the local error covariances evolve independently at each subsystem they may lose any coherence with the centralized error covariance. For example, in the estimation scheme in [14], the coupled states are applied as inputs to the local observers, but, the error covariances remain decoupled and no structure of the centralized error covariance is retained by the local filters. To keep coherence between the local covariances and the centralized covariance, we employ a *cooperative assimilation procedure* among the local error covariances that is based on approximating the centralized error process by a low dimensional Gauss–Markov error process.[1] The assimilation procedure is carried out with a distributed iterate-collapse inversion [(DICI), pronounced die-see] algorithm, briefly introduced in [19].

4) The DICI Algorithm and Information Filters: We implement the Kalman filter in the Information filter format [11], [20], which propagates in time the information matrices (inverse of the error covariances). The information matrices are inverted by the DICI algorithm in a distributed manner. Iterative matrix inversion can also be implemented using the distributed Jacobi algorithm [21], but the computational complexity of the distributed Jacobi scales linearly with the dimension n of the overall system, whereas the computational complexity of the DICI algorithm is independent of n, without compromising the convergence rate. In fact, the error process of the DICI algorithm is bounded above by the error process of the distributed Jacobi algorithm. We show the convergence of the iterate step of the DICI algorithm analytically and resort to numerical simulations to show the convergence of its collapse step.

In summary, spatial decomposition of complex large-scale systems, fusion algorithms for fusing observations, and the DICI algorithm to assimilate the local error covariances combine to give a robust, scalable, and distributed implementation of the Kalman filter.

We describe the rest of the paper. Section II motivates the discrete-time models, describes the centralized Information filters (CIFs) and the centralized L-banded Information filters (CLBIFs). Section III covers the model distribution step. We introduce the local Information filters in Section III-B along with the necessary notation. Section IV gives the observation fusion step of the local Information filters, and Section V presents the DICI algorithm. The filter step of the local Information filters is in Section VI, and the prediction step of the local Information filters is in Section VII. We conclude the paper with results in Section VIII and conclusions in Section IX. Appendix A discusses the L-banded inversion theorem, [17].

II. Background

In this section, we motivate the type of applications and large-scale dynamical systems of interest to us. The context is that of a time-varying random field governed by partial differential equations (PDEs); these systems can also be generalized to arbitrary dynamical systems belonging to a particular structural class, as we elaborate in Section II-A. To fix notation, we then present the centralized version of the Information filter.

A. Global Model

1) Global Dynamical System: Our goal here is to motivate how discrete linear models occur that exhibit a sparse and localized structure that we use to distribute the model in Section III. Examples include physical phenomena [22]–[26], e.g., ocean/wind circulation and heat/propagation equations, that can be broadly characterized by a PDE of the Navier–Stokes type. These are highly nonlinear and different regimens arise from different assumptions. For data assimilation, i.e., combining models with measured data, e.g., satellite altimetry data in ocean models, it is unfeasible to use nonlinear models; rather, linearized approximations (dynamical linearization) are employed. Hence, we take a very simplistic example and consider the discretization of a spatio–temporal dynamical system

$$\dot{x}_t = \mathcal{L}x_t + u_t \tag{1}$$

[1] In the error covariance domain, this approximation corresponds to the determinant/entropy maximizing completion of a partially specified (L-band, in our case) covariance matrix [15]–[18]. Such a completion results into a covariance matrix whose inverse is L-banded. We refer to a matrix as an L-banded matrix ($L \geq 0$), if the elements outside the band defined by the Lth upper and Lth lower diagonal are 0.

where $\dot{x}_t$ is the time partial derivative of a continuous-time physical phenomenon (e.g., heat, wave or wind), u_t is random noise, and $\mathcal{L}$, for example, is a second-order elliptical operator (that arises in the heat equation in diffusion)

$$\mathcal{L} = \alpha \frac{\partial^2}{\partial \rho_x^2} + \beta \frac{\partial^2}{\partial \rho_y^2} \tag{2}$$

where ρ_x and ρ_y represent the horizontal and vertical dimensions, respectively, and α and β are constants pertinent to the specific application. We start by discretizing the elliptical operator(2), using a standard second-order difference approximation on an $M \times J$ uniform mesh grid

$$\begin{aligned}\frac{\partial^2 x_t}{\partial \rho_x^2} &\sim x_{i+1,j} - 2x_{i,j} + x_{i-1,j}\\ \frac{\partial^2 x_t}{\partial \rho_y^2} &\sim x_{i,j+1} - 2x_{i,j} + x_{i,j-1}\end{aligned} \tag{3}$$

where x_{ij} is the value of the random field, x_t, at the ijth location in the $M \times J$ grid. We collect the variables, $\{x_{ij}\}$, in a state vector, $\mathbf{x}$, by, for example, using lexicographic ordering. Let $\mathbf{A}$ be a tridiagonal matrix, with zeros on the main diagonal and ones on the upper and lower diagonal; approximating the time derivative in (1) by using the forward Euler method, we can write the spatio–temporal discretization of (1) as

$$\mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{F}_c)\,\mathbf{x}_k + \mathbf{G}\mathbf{u}_k, \quad k \geq 0 \tag{4}$$

where k is the discrete-time index and the matrix $\mathbf{F}_c$ is given by

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{B} & \mathbf{C} & \\ \ddots & \ddots & \ddots \\ & \mathbf{C} & \mathbf{B} \end{bmatrix} = \mathbf{I} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} \tag{5}$$

where $\mathbf{B} = \mu\mathbf{I} + \beta_h\mathbf{A}$, and $\mathbf{C} = \beta_v\mathbf{I}$, the constants μ, β_h, and β_v are in terms of α, β in (2), and $\otimes$ is the Kronecker product [27]. Putting $\mathbf{F} = \mathbf{I} + \mathbf{F}_c$, the discrete-time dynamical system takes the form

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k. \tag{6}$$

In the above model, $\mathbf{x}_0 \in \mathbb{R}^n$ are the state initial conditions, $\mathbf{F} \in \mathbb{R}^{n \times n}$ is the model matrix, $\mathbf{u}_k \in \mathbb{R}^{j'}$ is the state noise vector and $\mathbf{G} \in \mathbb{R}^{n \times j'}$ is the state noise matrix.

Remarks: Here, we note that the model matrix, $\mathbf{F}$, is highly sparse, since the matrices $\mathbf{B}$ and $\mathbf{C}$ are at most tridiagonal and is perfectly banded in case of PDEs. We can relax this to sparse and localized matrices as when the coupling among the states decays with distance (in an appropriate measure); for example, see the spatially distributed systems in [28]. We mention briefly two other examples where such discrete-space-time models (with sparse and localized structure) also occur. In image processing, the dynamics at a pixel depends on neighboring pixel values [29], [30]; power grid models, under certain assumptions, exhibit banded structures [31]–[33]. As a final comment, systems that are sparse but not localized can be converted to sparse and localized by using matrix bandwidth reduction algorithms [34].

2) Observation Model: Let the system described in (6) be monitored by a network of N sensors. Observations at sensor l and time k are

$$\mathbf{y}_k^{(l)} = \mathbf{H}_l\mathbf{x}_k + \mathbf{w}_k^{(l)} \tag{7}$$

where $\mathbf{H}_l \in \mathbb{R}^{p_l \times n}$ is the local observation matrix for sensor l, p_l is the number of simultaneous observations made by sensor l at time k, and $\mathbf{w}_k^{(l)} \in \mathbb{R}^{p_l}$ is the local observation noise. In the context of the systems we are interested in, it is natural to assume that the observations are localized. These local observations at sensor l may be, e.g., the temperature or height at location l or an average of the temperatures or heights at l and neighboring locations. Mathematically, this can be characterized by assuming that $\mathbf{H}\mathbf{H}^T$ is sparse and banded, where $\mathbf{H}$ is the global observation matrix, introduced below.

We stack the observations at all N sensors in the sensor network to get the global observation model as follows. Let p be the total number of observations at all the sensors. Let the global observation vector, $\mathbf{y}_k \in \mathbb{R}^p$, the global observation matrix, $\mathbf{H} \in \mathbb{R}^{p \times n}$, and the global observation noise vector $\mathbf{w}_k \in \mathbb{R}^p$ be

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_k^{(1)} \\ \vdots \\ \mathbf{y}_k^{(N)} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_N \end{bmatrix}, \quad \mathbf{w}_k = \begin{bmatrix} \mathbf{w}_k^{(1)} \\ \vdots \\ \mathbf{w}_k^{(N)} \end{bmatrix}. \tag{8}$$

Then the global observation model is given by

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k. \tag{9}$$

We further assume that the overall system (6) and (9) is coupled, irreducible, and the pair $(\mathbf{F}, \mathbf{H})$ is observable.

3) Statistical Assumptions: We adopt standard assumptions on the statistical characteristics of the noise. The state noise sequence $\{\mathbf{u}_k\}_{k\geq 0}$, the observation noise sequence $\{\mathbf{w}_k\}_{k\geq 0}$, and the initial conditions $\mathbf{x}_0$ are independent, Gaussian, zero-mean, with

$$\begin{aligned}\mathbb{E}\left[\mathbf{u}_\iota\mathbf{u}_\tau^H\right] &= \mathbf{Q}\delta_{\iota\tau}, \text{ and}\\ \mathbb{E}\left[\mathbf{w}_\iota\mathbf{w}_\tau^H\right] &= \mathbf{R}\delta_{\iota\tau}, \text{ and}\\ \mathbb{E}\left[\mathbf{x}_0\mathbf{x}_0^H\right] &= \mathbf{S}_0\end{aligned} \tag{10}$$

where the superscript H denotes the Hermitian, the Kronecker delta $\delta_{\iota\tau} = 1$, if and only if $\iota = \tau$, and zero otherwise. Since the observation noises at different sensors are independent, we can partition the global observation noise covariance matrix, $\mathbf{R}$, into blocks $\mathbf{R}_l \in \mathbb{R}^{p_l \times p_l}$ corresponding to the local observation noise covariance matrices at each sensor l, as

$$\mathbf{R} = \text{blockdiag}[\mathbf{R}_1, \ldots, \mathbf{R}_N]. \tag{11}$$

For the rest of the presentation, we consider time-invariant models, specifically, the matrices $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{Q}$, and $\mathbf{R}$ are time invariant. The discussion, however, is not limited to either zero-mean initial conditions or time-invariant models and generalizations to the time-variant models will be added as we proceed.

B. Centralized Information Filter

Let $\breve{\mathbf{S}}_{k|k}$ and $\breve{\mathbf{S}}_{k|k-1}$ be the (filtered and prediction, respectively) error covariances, and their inverses be the information matrices $\breve{\mathbf{Z}}_{k|k}$ and $\breve{\mathbf{Z}}_{k|k-1}$. Let $\widehat{\mathbf{x}}_{k|k}$ and $\widehat{\mathbf{x}}_{k|k-1}$ be the filtered estimate and the predicted estimate of the state vector $\mathbf{x}_k$, respectively. We have the following relations:

$$\breve{\mathbf{S}}_{k|k} = \breve{\mathbf{Z}}_{k|k}^{-1} \tag{12}$$

$$\breve{\mathbf{S}}_{k|k-1} = \breve{\mathbf{Z}}_{k|k-1}^{-1}. \tag{13}$$

Define the n-dimensional global transformed state vectors as

$$\widehat{\mathbf{z}}_{k|k-1} = \breve{\mathbf{Z}}_{k|k-1}\widehat{\mathbf{x}}_{k|k-1} \tag{14}$$

$$\widehat{\mathbf{z}}_{k|k} = \breve{\mathbf{Z}}_{k|k}\widehat{\mathbf{x}}_{k|k}. \tag{15}$$

Define the n-dimensional global observation variables as

$$\mathbf{i}_k = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}_k \tag{16}$$

$$\mathcal{I} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \tag{17}$$

and the n-dimensional local observation variables at sensor l as

$$\mathbf{i}_{l,k} = \mathbf{H}_l^T\mathbf{R}_l^{-1}\mathbf{y}_k^{(l)} \tag{18}$$

$$\mathcal{I}_l = \mathbf{H}_l^T\mathbf{R}_l^{-1}\mathbf{H}_l. \tag{19}$$

When the observations are distributed among the sensors, see (7), the centralized information filter (CIF) can be implemented by collecting all the sensor observations at a central location; or, with observation fusion, by realizing that the global observation variables in (16)–(17) can be written as (see [3], [7], and [11])

$$\mathbf{i}_k = \sum_{l=1}^{N}\mathbf{i}_{l,k} \tag{20}$$

$$\mathcal{I} = \sum_{l=1}^{N}\mathcal{I}_l. \tag{21}$$

The *filter step* of the CIF is

$$\breve{\mathbf{Z}}_{k|k} = \breve{\mathbf{Z}}_{k|k-1} + \sum_{l=1}^{N}\mathcal{I}_l, \quad k \geq 0 \tag{22a}$$

$$\widehat{\mathbf{z}}_{k|k} = \widehat{\mathbf{z}}_{k|k-1} + \sum_{l=1}^{N}\mathbf{i}_{l,k}, \quad k \geq 0. \tag{22b}$$

The *prediction step* of the CIF is

$$\breve{\mathbf{Z}}_{k|k-1} = \breve{\mathbf{S}}_{k|k-1}^{-1} = \left(\mathbf{F}\breve{\mathbf{Z}}_{k-1|k-1}^{-1}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T\right)^{-1}, \quad k \geq 1, \quad \breve{\mathbf{Z}}_{0|-1} = \breve{\mathbf{S}}_0^{-1} \tag{23a}$$

$$\widehat{\mathbf{z}}_{k|k-1} = \breve{\mathbf{Z}}_{k|k-1}\left(\mathbf{F}\breve{\mathbf{Z}}_{k-1|k-1}^{-1}\widehat{\mathbf{z}}_{k-1|k-1}\right), \quad k \geq 1, \quad \widehat{\mathbf{z}}_{0|-1} = \mathbf{0}. \tag{23b}$$

The CIF needs i) the knowledge of all the observations, $\mathbf{y}_k$, at a central location to compute (20), a nontrivial communication task when the number of sensors N is large; and ii) global filter computations, e.g., (23), an infeasible challenge when the number of states n is very large. Further, the CIF has the disadvantages of large latency and a single point of failure.

C. Centralized L-Banded Information Filters

To avoid the $O(n^3)$ computations of the global quantities in (23), e.g., the inversion, $\breve{\mathbf{Z}}_{k-1|k-1}^{-1}$, we may approximate the information matrices $\breve{\mathbf{Z}}_{k|k}$ and $\breve{\mathbf{Z}}_{k|k-1}$, to be L-banded matrices, $\mathbf{Z}_{k|k}$ and $\mathbf{Z}_{k|k-1}$. We refer to the CIF with this approximation as the centralized L-banded Information filter (CLBIF). This approach is studied in [35], where the information loss between $\breve{\mathbf{Z}}$ and $\mathbf{Z}$, is given by the divergence

$$\begin{aligned}\mathrm{Div}_k(\breve{\mathbf{Z}}_{k|q}, \mathbf{Z}_{k|q}) &= \frac{1}{2}\left\|\breve{\mathbf{Z}}_{k|q}^{-T/2}\left(\mathbf{Z}_{k|q} - \breve{\mathbf{Z}}_{k|q}\right)\mathbf{Z}_{k|q}^{-1/2}\right\|_F^2, \\ &\leq \frac{1}{2}\left(\sum_i \lambda_{i(\mathbf{Z}_{k|q})}^{-1/2}\right)^2\left(\sum_i \lambda_{i(\breve{\mathbf{Z}}_{k|q})}^{-1/2}\right)^2\left\|\mathbf{Z}_{k|q} - \breve{\mathbf{Z}}_{k|q}\right\|_F^2\end{aligned} \tag{24}$$

where $q = k$ for estimation and $q = k - 1$ for prediction, $\|\cdot\|_F$ is the Frobenius norm, and $\lambda_{i(\mathbf{Z}_{k|q})}$ is the ith eigenvalue of the matrix $\mathbf{Z}_{k|q}$. Although, for a fixed k the divergence in (24) is bounded, the sequence $\mathrm{Div}_k(\cdot,\cdot)$ may be unbounded for small values of L. Here, we assume that L is chosen large enough, i.e., $L \geq L_{\min}$, such that $\lim_{k\to\infty}\mathrm{Div}_k(\cdot,\cdot)$ exists. The choice of $L_{\min}$ varies for different dynamical systems and loosely speaking depends on the structure of the model matrices $\mathbf{F}$.

This banded approximation of the information matrices is equivalent to the determinant/entropy maximizing completion of its inverse, a covariance matrix, part of whose elements are unspecified. In our case, the unspecified elements are the non-L -band elements, and it is well known that such completion of the covariance matrices have banded inverses with the same bandwidth; see, for instance, [15], [17], and [18], and the references within. Furthermore, such covariance matrices result from approximating the Gaussian error processes

$$\boldsymbol{\epsilon}_{k|k} = \mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}, \quad \boldsymbol{\epsilon}_{k|k-1} = \mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1} \tag{25}$$

to Gauss–Markov of the Lth order [16] (for $L = 1$, this has also been studied in [36]). Reference [17] presents an algorithm to derive the approximation that is optimal in Kullback–Leibler or maximum entropy sense in the class of all L-banded matrices approximating the inverse of the error covariance matrix. In the sequel, we assume this optimal L-banded approximation. The CLBIF (with the L-banded information matrices $\mathbf{Z}_{k|k}$ and $\mathbf{Z}_{k|k-1}$) is given by the *filter step* in (22a)–(22b) and the *prediction step* in (23a)–(23b), where the optimal information matrices $\breve{\mathbf{Z}}_{k|k}$ and $\breve{\mathbf{Z}}_{k|k-1}$ are replaced by their L-banded approximations. The algorithms in [17] and [37] reduce the computational complexity of the CLBIF to $O(n^2)$, but the resulting algorithm is still centralized and deals with the n-dimensional state. To distribute the CLBIF, we start by distributing the global model (6)–(9) in Section III.

III. Spatial Decomposition of Complex Large-Scale Systems

Instead of implementing CLBIF based on the global model, we implement local Information filters at the subsystems obtained by spatially decomposing the overall system. Section III-A deals with this decomposition by exploiting the sparse and localized structure of the model matrix $\mathbf{F}$.

A. Reduced Models at Each Sensor

This subsection shows how to distribute the global model (6) and (9), in order to get the reduced order subsystems. We illustrate the procedure with a simple example that reflects our assumptions on the dynamical system structure. Consider a five-dimensional system with the global dynamical model

$$\begin{aligned}\mathbf{x}_{k+1} &= \begin{bmatrix} f_{11} & f_{12} & 0 & 0 & 0\\ f_{21} & f_{22} & 0 & f_{24} & 0\\ f_{31} & 0 & f_{33} & 0 & 0\\ 0 & 0 & f_{43} & 0 & f_{45}\\ 0 & 0 & 0 & f_{54} & f_{55}\end{bmatrix}\mathbf{x}_k + \begin{bmatrix} 0 & 0\\ 0 & 0\\ 0 & g_{32}\\ 0 & 0\\ g_{51} & 0\end{bmatrix}\mathbf{u}_k\\ &= \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k. \end{aligned} \tag{26}$$

The system has two external noise sources $\mathbf{u}_k = [u_{1k}, u_{2k}]^T$. We monitor this system with $N = 3$ sensors, having scalar observations $y_k^{(l)}$ at each sensor l. The global observation vector $\mathbf{y}_k$ stacks the local observations, $y_k^{(l)}$, and is

$$\begin{aligned}\mathbf{y}_k = \begin{bmatrix} y_k^{(1)}\\ y_k^{(2)}\\ y_k^{(3)}\end{bmatrix} &= \begin{bmatrix} h_{11} & h_{12} & h_{13} & 0 & 0\\ 0 & h_{22} & h_{23} & h_{24} & 0\\ 0 & 0 & 0 & h_{34} & h_{35}\end{bmatrix}\mathbf{x}_k\\ &+ \begin{bmatrix} w_k^{(1)}\\ w_k^{(2)}\\ w_k^{(3)}\end{bmatrix} = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \end{aligned} \tag{27}$$

where $\mathbf{H} = [\mathbf{H}_1^T \quad \mathbf{H}_2^T \quad \mathbf{H}_3^T]^T$; the elements $\{f_{ij}\}$ and $\{h_{ij}\}$ in (26) and (27) are such that the dynamical system is $(\mathbf{F}, \mathbf{H})$-observable. We distribute the global model of (26) and (27) in Sections III-A-1) and III-A-2).

1) Graphical Representation Using System Digraphs: A system digraph visualizes the dynamical interdependence of the system. A system digraph [14], $\mathcal{J} = \{V, E\}$, is a directed graphical representation of the system, where $V = X \cup U$ is the vertex set consisting of the states $X = \{x_i\}_{i=1,\dots,n}$ and the noise inputs $U = \{u_i\}_{i=1,\dots,j'}$. The interconnection matrix E is the binary representation (having a 1 for each nonzero entry) of the model matrix $\mathbf{F}$ and the state noise matrix $\mathbf{G}$ concatenated together. The interconnection matrix E for the system in (26) is

$$E = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 1 & 1 & 0\end{bmatrix}. \tag{28}$$

The system digraph is shown in Fig. 1(a).

2) Subsystem Derivation Using Cut-Point Sets: We have $N = 3$ sensors monitoring the system through the observation model (27). We implement a subsystem at each sensor; thus, subsystem l corresponds to sensor l. We associate to each subsystem l a *cut-point set* $V^{(l)}$, where $V^{(l)} \subseteq X$. We choose to include the states in a cut-point set that are observed by the sensors in its corresponding subsystem; see [38] for an alternate definition of the cut-point sets, and algorithms to find all cut-point sets and a minimal cut-point set, if it exists. The cut-point sets select the local states involved in the local dynamics at each subsystem. From (27), the cut-point sets[2] are shown in Fig. 1(b), where we have the following cut-point set, e.g., at subsystem 1, $V^{(1)} = \{x_1, x_2, x_3\}$.

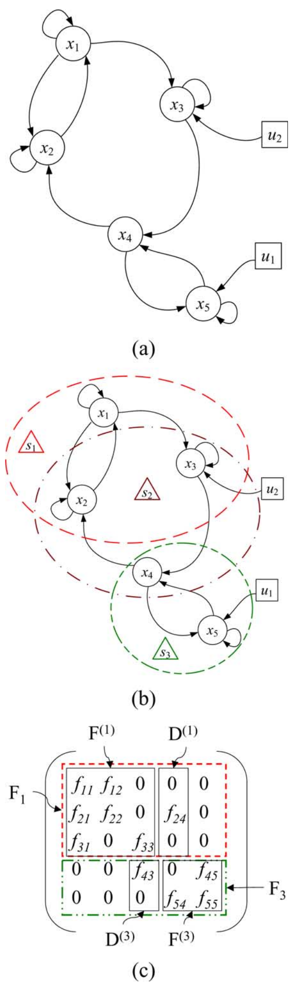


Fig. 1. System digraph and cut-point sets: (a) Digraph representation of the 5-D system (26)–(27). The circles represent the states $\mathbf{x}$, and the squares represent the input noise sources $\mathbf{u}$. (b) Cut-point sets associated to the three subsystems ($\triangle$) are shown by the dashed circles. (c) Partitioning of the global model matrix $\mathbf{F}$ into local model matrices $\mathbf{F}^{(l)}$, and the local internal input matrices $\mathbf{D}^{(l)}$, shown for subsystem 1 and subsystem 3, from the example system (26)–(27).

3) Dimension of the Subsystems: The local states at subsystem l, i.e., the components of its local state vector $\mathbf{x}_k^{(l)}$ are the elements in its associated cut-point set $V^{(l)}$. The dimension of the local Kalman filter implemented at subsystem l is now n_l. The set of n_l-dimensional local Kalman filters will give rise, as

[2] For simplicity of the presentation, we chose here that each state variable is observed at least by one subsystem. We can easily account for this when this is not true by extending the cut-point sets $V^{(l)}$ to $\overline{V}^{(l)}$, such that $\bigcup_{l=1}^{N} \overline{V}^{(l)} = X$.

will be clear later, to an L-banded centralized information matrix with $L = \min(n_1, \ldots, n_N)$. The loss in the optimality as a function of L is given by the divergence (24). Hence, for a desired level of performance, i.e., for a fixed L, we may need to extend (include additional states in a cut-point set) the cut-point sets $V^{(l)}$ to $V_L^{(l)}$, such that

$$n_l = \left| V_L^{(l)} \right| \geq L \quad \forall\, l \tag{29}$$

where $|\cdot|$ when applied to a set denotes its cardinality. This procedure of choosing an L based on a certain desired performance gives a lower bound on the dimension of each subsystem.

4) Coupled States as Inputs: The directed edges coming into a cut-point set are the inputs required by the local model at that subsystem. In the context of our running illustration (26)–(27), we see that the local state vector for subsystem 1 is $\mathbf{x}_k^{(1)} = [x_{1,k}, x_{2,k}, x_{3,k}]^T$, and the inputs to the local model consist of a subset of the state set X (at subsystem 1, $x_{4,k}$ is the input coming from subsystem 2) and a subset of the noise input set U ($u_{2,k}$ at subsystem s_1).

5) Local Models: For the local model at subsystem l, we collect the states required as input in a local internal input vector $\mathbf{d}_k^{(l)}$ (we use the word *internal* to distinguish from the externally applied inputs), and the noise sources required as input in a local noise input vector $\mathbf{u}_k^{(l)}$. We collect the elements from $\mathbf{F}$ corresponding to the local state vector $\mathbf{x}_k^{(l)}$ in a local model matrix $\mathbf{F}^{(l)}$. Similarly, we collect the elements from $\mathbf{F}$ corresponding to the local internal input vector $\mathbf{d}_k^{(l)}$ in a local internal input matrix $\mathbf{D}^{(l)}$, and the elements from $\mathbf{G}$ corresponding to the local noise input vector $\mathbf{u}_k^{(l)}$ in a local state noise matrix $\mathbf{G}^{(l)}$. Fig. 1(c) shows this partitioning for subsystems 1 and 3. We have the following local models from (26):

$$\begin{aligned}
\mathbf{x}_{k+1}^{(1)} &= \begin{bmatrix} f_{11} & f_{12} & 0 \\ f_{21} & f_{22} & 0 \\ f_{31} & 0 & f_{33} \end{bmatrix} \mathbf{x}_k^{(1)} \\
&\quad + \begin{bmatrix} 0 \\ f_{24} \\ 0 \end{bmatrix} x_{4,k} + \begin{bmatrix} 0 \\ 0 \\ g_{32} \end{bmatrix} u_{2,k}, \\
&= \mathbf{F}^{(1)}\mathbf{x}_k^{(1)} + \mathbf{D}^{(1)}\mathbf{d}_k^{(1)} + \mathbf{G}^{(1)}\mathbf{u}_k^{(1)} \qquad (30) \\
\mathbf{x}_{k+1}^{(2)} &= \begin{bmatrix} f_{22} & 0 & f_{42} \\ 0 & f_{33} & 0 \\ 0 & f_{43} & 0 \end{bmatrix} \mathbf{x}_k^{(2)} \\
&\quad + \begin{bmatrix} f_{21} & 0 \\ f_{31} & 0 \\ 0 & f_{45} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{5,k} \end{bmatrix} + \begin{bmatrix} 0 \\ g_{32} \\ 0 \end{bmatrix} u_{2,k} \\
&= \mathbf{F}^{(2)}\mathbf{x}_k^{(2)} + \mathbf{D}^{(2)}\mathbf{d}_k^{(2)} + \mathbf{G}^{(2)}\mathbf{u}_k^{(2)} \qquad (31) \\
\mathbf{x}_{k+1}^{(3)} &= \begin{bmatrix} 0 & f_{45} \\ f_{54} & f_{55} \end{bmatrix} \mathbf{x}_k^{(3)} + \begin{bmatrix} f_{43} \\ 0 \end{bmatrix} x_{3,k} + \begin{bmatrix} 0 \\ g_{51} \end{bmatrix} u_{1,k} \\
&= \mathbf{F}^{(3)}\mathbf{x}_k^{(3)} + \mathbf{D}^{(3)}\mathbf{d}_k^{(3)} + \mathbf{G}^{(3)}\mathbf{u}_k^{(3)}. \qquad (32)
\end{aligned}$$

We may also capture the above extraction of the local states by the cut-point sets, with the following procedure. Let the total number of states in the cut-point set at subsystem l, $V^{(l)}$, be n_l. Let $\mathbf{T}_l$ be an $n_l \times n$ selection matrix, such that it selects n_l states in the cut-point set $V^{(l)}$ from the entire state vector $\mathbf{x}_k$, according to the following relation:

$$\mathbf{x}_k^{(l)} = \mathbf{T}_l \mathbf{x}_k. \tag{33}$$

For example, the selection matrix $\mathbf{T}_1$ at subsystem 1 is

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{34}$$

We establish a *reduced* local observation matrix $\mathbf{H}^{(l)}$ by retaining the terms corresponding to the local state vector $\mathbf{x}_k^{(l)}$ from the local observation matrix $\mathbf{H}_l$. We may write

$$\mathbf{H}^{(l)} = \mathbf{H}_l \mathbf{T}_l^{\#} \tag{35}$$

where "#" denotes the pseudo-inverse of the matrix. In the context of our running illustration, the reduced local observation matrix $\mathbf{H}^{(1)} = [1, 1, 1]$ is obtained from the local observation matrix $\mathbf{H}_1 = [1, 1, 1, 0, 0]$. Note that $\mathbf{H}_l$ picks the states from the global state vector $\mathbf{x}_k$, whereas $\mathbf{H}^{(l)}$ picks the states from the local state vector $\mathbf{x}_k^{(l)}$. The reduced local observation models are given by

$$\mathbf{y}_k^{(l)} = \mathbf{H}^{(l)}\mathbf{x}_k^{(l)} + \mathbf{w}_k^{(l)}. \tag{36}$$

We now make some additional comments. For simplicity of the explanation, we refer to our running example (26)–(27). We note that the subsystems overlap, as shown by the overlapping cut-point sets in Fig. 1(b). Due to this overlap, observations corresponding to the shared states are available at multiple subsystems that should be fused. We further note that the local model (30) at subsystem 1 is coupled to the local model (31) at subsystem 2 through the state $x_{4,k}$. The state $x_{4,k}$ at subsystem 1 does not appear in the local state vector, i.e., $x_{4,k} \notin \mathbf{x}_k^{(1)}$. But, it is still required as an internal input at subsystem 1 to preserve the global dynamics. Hence, subsystem 2 communicates the state $x_{4,k}$, which appears in its local state vector, i.e., $x_{4,k} \in \mathbf{x}_k^{(2)}$, to subsystem 1. Hence, at an arbitrary subsystem l, we derive the reduced model to be

$$\mathbf{x}_{k+1}^{(l)} = \mathbf{F}^{(l)}\mathbf{x}_k^{(l)} + \mathbf{D}^{(l)}\mathbf{d}_k^{(l)} + \mathbf{G}^{(l)}\mathbf{u}_k^{(l)}. \tag{37}$$

Since the value of the state itself is unknown, subsystem 2 communicates its estimate $\widehat{x}_{4,k|k}^{(2)}$ to subsystem 1. This allows subsystem 1 to complete its local model and preserve global dynamics, thus, taking into account the coupling subsystems. This process is repeated at all subsystems. Hence, the local internal input vector $\mathbf{d}_k^{(l)}$ is replaced by its estimate $\widehat{\mathbf{d}}_{k|k}^{(l)}$. It is worth mentioning here that if the dynamics were time-dependent, i.e., the matrices, $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ change with time k, then the above decomposition procedure will have to be repeated at each k. This may result into a different communication topology over which the subsystems communicate at each k.

B. Local Information Filters

To distribute the estimation of the global state vector $\mathbf{x}_k$, we implement local Information filters (LIFs) at each subsystem l, which are based on the subsystem models (37) and (36). Each

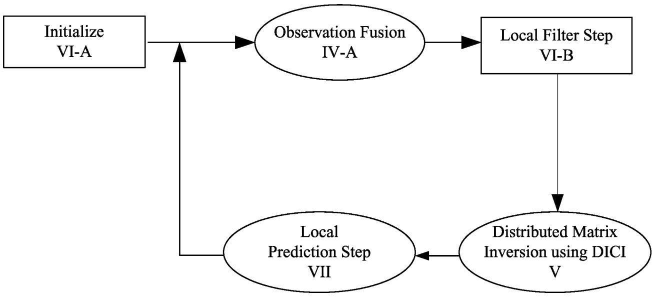


Fig. 2. Block Diagram for the LIFs: Steps involved in the LIF implementation. The ovals represent the steps that require local communication.

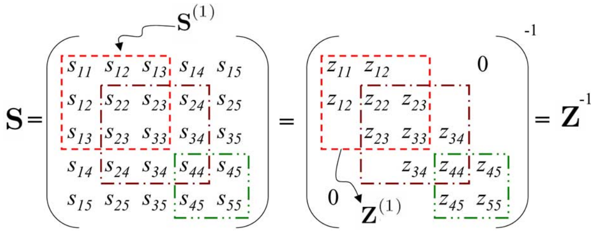

$\mathbf{S}^{(1)}$
$\mathbf{S}=$
s_{11} s_{12} s_{13} s_{14} s_{15}
s_{12} s_{22} s_{23} s_{24} s_{25}
s_{13} s_{23} s_{33} s_{34} s_{35}
s_{14} s_{24} s_{34} s_{44} s_{45}
s_{15} s_{25} s_{35} s_{45} s_{55}
=
z_{11} z_{12}
0
z_{12} z_{22} z_{23}
z_{23} z_{33} z_{34}
z_{34} z_{44} z_{45}
0
z_{45} z_{55}
-1
$\mathbf{Z}^{(1)}$
$= \mathbf{Z}^{-1}$

Fig. 3. Relationship between the global error covariance matrices $\mathbf{S}$ and their inverses, the global information matrices $\mathbf{Z}$, with $L = 1$-banded approximation on $\mathbf{Z}$. The figure also shows how the local matrices $\mathbf{S}^{(l)}$ and $\mathbf{Z}^{(l)}$ constitute their global counterparts. Since this relation holds for both the estimation and prediction matrices, we remove the subscripts.

LIF computes local quantities (matrices and vectors of dimension n_l), which are then fused (if required) by exchanging information among the neighbors. Some of the update procedures are iterative. Although, no centralized knowledge of the estimation of the global state exists, the union of the local state vector represents, in a distributed way, the knowledge that exists in a centralized fashion in the CLBIF. In most applications, nowhere in the network is there the need for this centralized knowledge.

The LIFs consist of initial conditions, a local filter step (including observation fusion and distributed matrix inversion) and a local prediction step; see Fig. 2. These steps are presented in the next four sections. To proceed with Sections IV–VIII, we provide next needed notation.

Notation: The superscript (l) refers to a local reduced-order variable ($n_l \times 1$ vector or $n_l \times n_l$ matrix) at subsystem l. For example, the reduced observation vector, $\mathbf{i}_k^{(l)}$, and the reduced observation matrix, $\mathcal{I}^{(l)}$, are

$$\mathbf{i}_k^{(l)} = (\mathbf{H}^{(l)})^T \mathbf{R}_l^{-1} \mathbf{y}_k^{(l)} \tag{38}$$
$$\mathcal{I}^{(l)} = (\mathbf{H}^{(l)})^T \mathbf{R}_l^{-1} \mathbf{H}^{(l)}. \tag{39}$$

The local error covariance matrices $\mathbf{S}_{k|k}^{(l)}$ and $\mathbf{S}_{k|k-1}^{(l)}$ are the overlapping diagonal submatrices of the global error covariance matrices $\mathbf{S}_{k|k}$ and $\mathbf{S}_{k|k-1}$. Let $\mathbf{Z}_{k|k}^{(l)}$ and $\mathbf{Z}_{k|k-1}^{(l)}$ be the local information matrices. These local information matrices are overlapping diagonal submatrices of the global L-banded information matrices $\mathbf{Z}_{k|k}$ and $\mathbf{Z}_{k|k-1}$. These local matrices overlap because the subsystems overlap. Fig. 3 captures the relationship between the local error covariance matrices and the local information matrices given by (12) and (13).

IV. Overlapping Reduced Models

After the spatial decomposition of the dynamical system, introduced in Section III, the resulting subsystems share state variables, as shown by the overlapped cut-point sets in Fig. 1(b). Since the subsystems sharing the states have (conditionally) independent observations of the shared states, observations corresponding to the shared states should be fused. We present observation fusion in Section IV-A with the help of bipartite fusion graphs, [12].

A. Observation Fusion

Equations (20) and (21) show that the observation fusion is equivalent to adding the corresponding n-dimensional local observation variables (18)–(19). In CLBIF, we implement this fusion directly because each local observation variable in (18)–(19) corresponds to the full n-dimensional state vector $\mathbf{x}_k$. Since the n_l-dimensional *reduced* observation variables (38)–(39), correspond to different local state vectors, $\mathbf{x}_k^{(l)}$, they cannot be added directly.

For simplicity and without loss of generality, we assume each local observation matrix $\mathbf{H}_i$ to be a row. To achieve observation fusion, we introduce the following undirected bipartite fusion graph[3] $\mathcal{B}$. Let $S_N = \{s_1, \ldots, s_N\}$ be the set of sensors and X be the set of states. The vertex set of the bipartite fusion graph $\mathcal{B}$ is $S_N \cup X$. We now define the edge set $E_{\mathcal{B}}$ of the fusion graph $\mathcal{B}$. The sensor s_i is connected to the state variable x_j, if s_i observes (directly or as a linear combination) the state variable x_j. In other words, we have an edge between sensor s_i and state variable x_j, if the local observation matrix $\mathbf{H}_i$ at sensor s_i contains a nonzero entry in its jth column. Fig. 4(a) shows the bipartite graph for the example system in (26)–(27). The set of sensors S_j that observe the jth state x_j come directly from the bipartite fusion graph $\mathcal{B}$. For example, from Fig. 4(a), we see that S_1 contains s_1 as a single sensor, whereas S_2 contains the sensors s_1, s_2, and so on.[4] States having more than one sensor connected to them in the bipartite fusion graph $\mathcal{B}$ are the states for which fusion is required, since we have multiple observations for that state.

With the help of the above discussion, we establish the fusion of the reduced observation variables, (38)–(39). The reduced model at each sensor involves n_l state variables, and each element in the $n_l \times 1$ reduced observation vector $\mathbf{i}_k^{(l)}$ corresponds to one of these states, i.e., each entry in $\mathbf{i}_k^{(l)}$ has some information about its corresponding state variable. Let the entries of the $n_l \times 1$ reduced observation vector $\mathbf{i}_k^{(l)}$ at sensor l be subscripted

[3] A bipartite graph is a graph whose vertices can be divided into two disjoint sets X and S_N, such that every edge connects a vertex in X to a vertex in S_N, and there is no edge between any two vertices of the same set [39].

[4] Mathematically, this can be captured as follows. Let the number of nonzero elements in the jth column, $\mathbf{h}_j$, of the global observation matrix $\mathbf{H}$ be given by N_j and let Ω_j be the set of the locations of these nonzero elements, with $|\Omega_j| = N_j$ and $\Omega_{j,\varsigma}$ be its ςth element. Let $\mathbf{V}_j$ be an $N_j \times N$ matrix with ones on the $(\varsigma, \Omega_{j,\varsigma})$th locations and zeros elsewhere. Considering S_j $\forall j$ as a column vector, we have $S_j = \mathbf{V}_j S_N$.

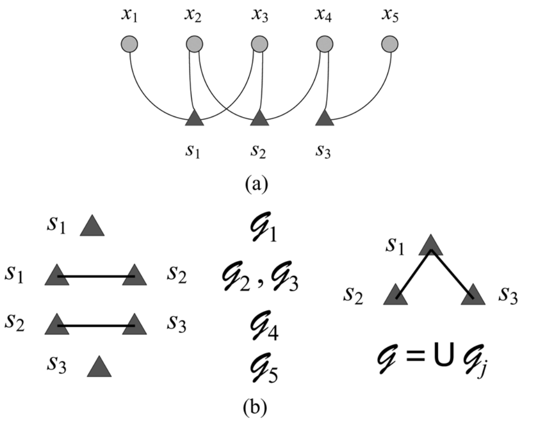


Fig. 4. (a) Bipartite Fusion graph, $\mathcal{B}$, is shown for the example system (26)–(27). (b) Subgraphs $\mathcal{G}_j$ for observation fusion.

by the n_l state variables modeled at sensor l. In the context of the example given by system (26)–(27), we have

$$\mathbf{i}_k^{(1)} = \begin{bmatrix} i_{k,x_1}^{(1)} \\ i_{k,x_2}^{(1)} \\ i_{k,x_3}^{(1)} \end{bmatrix}, \quad \mathbf{i}_k^{(2)} = \begin{bmatrix} i_{k,x_2}^{(2)} \\ i_{k,x_3}^{(2)} \\ i_{k,x_4}^{(2)} \end{bmatrix}, \quad \mathbf{i}_k^{(3)} = \begin{bmatrix} i_{k,x_4}^{(3)} \\ i_{k,x_5}^{(3)} \end{bmatrix}. \quad (40)$$

For the jth state x_j, the observation fusion is carried out on the set of sensors S_j attached to this state in the bipartite fusion graph $\mathcal{B}$. The fused observation vectors denoted by $\mathbf{i}_{f,k}^{(l)}$ are given by

$$\mathbf{i}_{f,k}^{(1)} = \begin{bmatrix} i_{k,x_1}^{(1)} \\ i_{k,x_2}^{(1)} + i_{k,x_2}^{(2)} \\ i_{k,x_3}^{(1)} + i_{k,x_3}^{(2)} \end{bmatrix}$$

$$\mathbf{i}_{f,k}^{(2)} = \begin{bmatrix} i_{k,x_2}^{(2)} + i_{k,x_2}^{(1)} \\ i_{k,x_3}^{(2)} + i_{k,x_3}^{(1)} \\ i_{k,x_4}^{(2)} + i_{k,x_4}^{(3)} \end{bmatrix}$$

$$\mathbf{i}_{f,k}^{(3)} = \begin{bmatrix} i_{k,x_4}^{(3)} + i_{k,x_4}^{(2)} \\ i_{k,x_5}^{(3)} \end{bmatrix}. \quad (41)$$

Generalizing to the arbitrary sensor l, we may write the entry, $i_{f,k,x_j}^{(l)}$, corresponding to x_j in the fused observation vector, $\mathbf{i}_{f,k}^{(l)}$ as

$$i_{f,k,x_j}^{(l)} = \sum_{s \in S_j} i_{k,x_j}^{(s)} \quad (42)$$

where $i_{k,x_j}^{(s)}$ is the entry corresponding to x_j in the reduced observation vector at sensor s, $\mathbf{i}_k^{(s)}$.

B. Implementation

We now provide further notation on the communication topology to formulate the observation fusion procedure precisely. For the jth state x_j, let $\mathcal{G}_j^{(fc)} = \left\{S_j, E_j^{(fc)}\right\}$ be a fully connected[5] (fc) subgraph, such that its vertices are all the sensors that observe x_j. With this definition, we can define $E_j^{(fc)} = \overline{\mathbf{V}_j\left(\mathbf{I} - \overline{\mathbf{h}_j\mathbf{h}_j^T}\right)\mathbf{V}_j^{\#}}$, where $\mathbf{V}_j$ is given in footnote 4 and the overline denotes the binary representation, as introduced in Section III-A-1). For the jth state x_j, let $\mathcal{G}_j^{(c)} = \left\{S_j, E_j^{(c)} \subseteq E_j^{(fc)}\right\}$ be a connected[6] subgraph. Now, the overall sensor network topology required for the observation fusion with fully connected subgraphs $\mathcal{G}_j^{(fc)}$ can be given by $\mathcal{G} = \left\{S_N, \overline{\sum_j \mathbf{V}_j^{\#} E_j^{(fc)} \mathbf{V}_j} = \overline{\mathbf{I} - \overline{\mathbf{HH}}^T}\right\}$ and for the connected subgraphs $\mathcal{G}_j^{(c)}$ can be given by $\widetilde{\mathcal{G}} = \left\{S_N, \overline{\sum_j \mathbf{V}_j^{\#} E_j^{(c)} \mathbf{V}_j}\right\}$. These graphs are shown in Fig. 4(b).

Remarks: We now make some comments. First, if we choose the overall sensor communication graph, $\mathcal{G}$, that results from the union of the fully-connected subgraphs $\mathcal{G}_j^{(fc)}$ ∀ j, the observation fusion procedure *does not* require an iterative procedure and is realized in a single step. If we choose the overall sensor communication graph $\widetilde{\mathcal{G}}$ that results from the union of the connected subgraphs $\mathcal{G}_j^{(c)}$ ∀ j, the observation fusion procedure requires an iterative consensus algorithm[7] and can be realized using weighted averaging algorithms [13].

With the assumption of a localized global observation matrix, $\mathbf{H}$, as motivated in Section II-A-2, the overall sensor communication network, $\mathcal{G}$ or $\widetilde{\mathcal{G}}$, is not necessarily fully connected, as shown in Fig. 4(b).

With any choice of the overall sensor communication graph $\mathcal{G}$ or $\widetilde{\mathcal{G}}$, the communication required for observation fusion is single-hop.

It is worth mentioning that the observation fusion procedure implemented in [7] cannot be realized in single-step unless the overall sensor communication graph $\mathcal{G}$ is fully connected, which we *do not* require anywhere in our solution.

A similar procedure on the pairs of state variables and their associated subgraphs can be implemented to fuse the reduced observation matrices $\mathcal{I}^{(l)}$. Since we assume the observation model to be stationary ($\mathbf{H}$ and $\mathbf{R}$ are time-independent), the fusion on the reduced observation matrix $\mathcal{I}^{(l)}$ is to be carried out only once and can be an offline procedure. If that is not the case, and $\mathbf{H}$ and $\mathbf{R}$ are time dependent, fusion on $\mathcal{I}$ has to be repeated at each time k.

A comment on estimate fusion. Since we fuse the observations concerning the shared states among the sensors, one may ask if it is required to carry out fusion of the estimates of the shared states. It turns out that consensus on the observations leads to consensus on the estimates. This will become clear with the introduction of the local filter and the local prediction step of the LIFs; therefore, we defer the discussion on estimate fusion to Section VII-C.

[5] A fully-connected graph, $\mathcal{G}^{(fc)} = \{S_N, E^{(fc)}\}$, is such that every pair of distinct vertices in S_N is connected by an edge in $E^{(fc)}$.

[6] A connected graph, $\mathcal{G}^{(c)} = \{S_N, E^{(c)}\}$, is such that there exists a path from any vertex in S_N to any other vertex in S_N.

[7] In this case, we assume that the communication is fast enough so that the consensus algorithm can converge, see [40] for a discussion on distributed Kalman filtering based on consensus strategies. The convergence of the consensus algorithm is shown to be geometric and the convergence rate can be increased by optimizing the weight matrix for the consensus iterations using semidefinite programming [41]. The communication topology of the sensor network can also be improved to increase the convergence speed of the consensus algorithms [42].

V. Distributed Matrix Inversion With Local Communication

In this section, we discuss the cooperative assimilation procedure on the local error covariances. Consider the example model (26)–(27), when we employ LIFs on the distributed models (30)–(32). The local estimation information matrices $\mathbf{Z}_{k|k}^{(1)}$, $\mathbf{Z}_{k|k}^{(2)}$, and $\mathbf{Z}_{k|k}^{(3)}$ correspond to the overlapping diagonal submatrices of the global 5×5 estimation information matrix, $\mathbf{Z}_{k|k}$; see Fig. 3, with $L = 1$-banded approximation on $\mathbf{Z}_{k|k}$. It will be shown (Section VII-A) that the local prediction information matrix $\mathbf{Z}_{k+1|k}^{(l)}$ is a function of the local error covariance matrices $\mathbf{S}_{k|k}^{(l)}$, and hence we need to compute $\mathbf{S}_{k|k}^{(l)}$ from the local filter information matrices $\mathbf{Z}_{k|k}^{(l)}$, which we get from the local filter step (Section VI). As can be seen from Fig. 3 and (12), for these local submatrices

$$\mathbf{S}^{(l)} \neq \left(\mathbf{Z}^{(l)}\right)^{-1}. \tag{43}$$

Collecting all the local information matrices $\mathbf{Z}_{k|k}^{(l)}$ at each sensor and then carrying out an $n \times n$ matrix inversion is *not* a practical solution for large-scale systems (where n may be large), because of the large communication overhead and $O(n^3)$ computational cost. Using the L-banded structure on the global estimation information matrix $\mathbf{Z}_{k|k}$, we present below a distributed iterate collapse inversion with overrelaxation [(DICI-OR), pronounced *die-see*-O-R] algorithm.[8] We present a generalization of the centralized Jacobi overrelaxation (JOR) algorithm to solve matrix inversion in Section V-A and show that the computations required in its distributed implementation scale linearly with the dimension, n, of the system. We then present the DICI-OR algorithm and show that it is independent of the dimension, n, of the system. For simplicity of the presentation, we ignore the Kalman filter time subscripts, k, in the remainder of this section.

A. *Centralized Jacobi Overrelaxation (JOR) Algorithm*

The centralized Jacobi overrelaxation (JOR) algorithm for vectors [21] solves a linear system of n equations iteratively, by successive substitution. It can be easily extended to get the centralized JOR algorithm for matrices that solves

$$\mathbf{ZS} = \mathbf{T} \tag{44}$$

for the unknown matrix $\mathbf{S}$, where the matrices $\mathbf{Z}$ and $\mathbf{T}$ are known. Let $\mathbf{M} = \operatorname{diag}(\mathbf{Z})$, for some $\gamma > 0$

$$\mathbf{S}_{t+1} = \left((1-\gamma)\mathbf{I} + \gamma\mathbf{M}^{-1}(\mathbf{M}-\mathbf{Z})\right)\mathbf{S}_t + \gamma\mathbf{M}^{-1}\mathbf{T} \tag{45}$$

converges to $\mathbf{S}$ and is the centralized JOR algorithm for matrices, solving n coupled linear systems of equations (44), where γ is sometimes called a relaxation parameter [21]. Putting $\mathbf{T} = \mathbf{I}$, we can iteratively solve for $\mathbf{ZS} = \mathbf{I} \Rightarrow \mathbf{S} = \mathbf{Z}^{-1}$, and, if $\mathbf{Z}$ is known, the following iterations converge to $\mathbf{Z}^{-1}$:

$$\mathbf{S}_{t+1} = \mathbf{P}_\gamma \mathbf{S}_t + \gamma\mathbf{M}^{-1} \tag{46}$$

where the multiplier matrix $\mathbf{P}_\gamma$ is defined as

$$\mathbf{P}_\gamma = (1-\gamma)\mathbf{I} + \gamma\mathbf{M}^{-1}(\mathbf{M}-\mathbf{Z}). \tag{47}$$

The Jacobi algorithm can now be considered as a special case of the JOR algorithm with $\gamma = 1$.

1) Convergence: Let $\mathbf{S}$ be the stationary point of the iterations in (46), and let $\widetilde{\mathbf{S}}_{t+1}$ denote the iterations in (46). It can be shown that the error process $\widetilde{\mathbf{E}}_{t+1} = \widetilde{\mathbf{S}}_{t+1} - \mathbf{S}$ for the JOR algorithm is

$$\widetilde{\mathbf{E}}_{t+1} = \mathbf{P}_\gamma \widetilde{\mathbf{E}}_t \tag{48}$$

which decays to zero if $\|\mathbf{P}_\gamma\|_2 < 1$, where $\|\cdot\|_2$ denotes the spectral norm of a matrix. The JOR algorithm (46) converges for all symmetric positive definite matrices, $\mathbf{Z}$, for sufficiently small $\gamma > 0$, see [21], an alternate convergence proof is provided in [46] via convex M-matrices, whereas convergence for parallel asynchronous team algorithms is provided in [47]. Since the information matrix $\mathbf{Z}$ is the inverse of an error covariance matrix; $\mathbf{Z}$ is symmetric positive definite by definition, and the JOR algorithm always converges. Plugging $\gamma = 1$ in (46) gives us the centralized Jacobi algorithm for matrices, which converges for all diagonally dominant matrices; see [21]. We can further write the error process as

$$\widetilde{\mathbf{E}}_{t+1} = \mathbf{P}_\gamma^{t+1}(\mathbf{S}_0 - \mathbf{S}) \tag{49}$$

where the matrix $\mathbf{S}_0$ is the initial condition. The spectral norm of the error process can be bounded by

$$\|\widetilde{\mathbf{E}}_{t+1}\|_2 \leq \max_i \left|\lambda_{i(\mathbf{P}_\gamma^H \mathbf{P}_\gamma)}\right|^{t+1/2} \|(\mathbf{S}_0 - \mathbf{S})\|_2. \tag{50}$$

The JOR algorithm is centralized as it requires the complete $n \times n$ matrices involved. This requires global communication and an nth order computation at each iteration of the algorithm. We present below its distributed implementation.

2) Distributed JOR Algorithm: We are interested in the local error covariances that lie on the L-band of the matrix $\mathbf{S} = \mathbf{Z}^{-1}$. Distributing the JOR (in addition to [21], distributed Jacobi and Gauss–Seidel type iterative algorithms can also be found in [43], [45], and [46]) algorithm (46) directly to compute the L-band of $\mathbf{S}$ gives us the following equations for the ijth element s_{ij} in $\mathbf{S}_{t+1}$

$$s_{ij,t+1} = \begin{cases} \mathbf{p}_i \mathbf{s}_t^j & i \neq j \\ \mathbf{p}_i \mathbf{s}_t^j + \gamma m_{ii}^{-1}, & i = j \end{cases} \tag{51}$$

where the row vector $\mathbf{p}_i$ is the ith row of the multiplier matrix $\mathbf{P}_\gamma$; the column vector $\mathbf{s}_t^j$ is the jth column of the matrix $\mathbf{S}_t$; and the scalar element m_{ii} is the ith diagonal element of the diagonal

[8] It is worth mentioning here that the DICI algorithm [for solving $\mathbf{ZS} = \mathbf{I}$, with (symmetric positive definite) SPD L-banded matrix $\mathbf{Z} \in \mathbb{R}^{n \times n}$ and the $n \times n$ identity matrix $\mathbf{I}$] is neither a direct extension nor a generalization of (block) Jacobi or Gauss–Seidel type iterative algorithms (that solve a vector version, $\mathbf{Zs} = \mathbf{b}$ with $\mathbf{s}, \mathbf{b} \in \mathbb{R}^n$, of $\mathbf{ZS} = \mathbf{I}$; see [21], [43]–[45]). Using the Jacobi or Gauss-Seidel type iterative schemes for solving $\mathbf{ZS} = \mathbf{I}$ is equivalent to solving n linear systems of equations, $\mathbf{Zs} = \mathbf{b}$; hence, the complexity scales linearly with n. Instead the DICI algorithm employs a nonlinear collapse operator that exploits the structure of the inverse, $\mathbf{S}$, of a SPD L-banded matrix, $\mathbf{Z}$, which makes its complexity independent of n.

matrix $\mathbf{M}$. Since the matrix $\mathbf{Z}$ is L-banded, the multiplier matrix $\mathbf{P}_\gamma$ in (47) is also L-banded. The ith row of the multiplier matrix $\mathbf{P}_\gamma$ contains nonzeros at most at $2L+1$ locations in the index set $\kappa = \{i-L, \ldots, i, \ldots, i+L\}$. These nonzero elements pick the corresponding elements with indices in the index set κ in the jth column $\mathbf{s}_t^j$ of $\mathbf{S}_t$. Due to the L-bandedness of the multiplier matrix $\mathbf{P}_\gamma$, the JOR algorithm can be easily distributed with appropriate communication with the neighboring sensors.

A major drawback of the distributed JOR algorithm is that at each sensor the computation requirements scale linearly with the dimension n of the system. This can be seen by writing out the iteration in (51), e.g., for an L-banded element such that $|i-j| = L$ (i.e., for the elements that lie on the Lth upper or Lth lower diagonal). In the context of Fig. 3, we can write $s_{45,t+1}$ from (51) as

$$s_{45,t+1} = p_{43}s_{35,t} + p_{44}s_{45,t} + p_{45}s_{55,t}. \tag{52}$$

The element $s_{35,t}$ does not lie in the L-band of $\mathbf{S}$, and, hence, does not belong to any local error covariance matrix $\mathbf{S}^{(l)}$. Iterating on it using (51) gives

$$s_{35,t+1} = p_{32}s_{25,t} + p_{33}s_{35,t} + p_{34}s_{45,t}. \tag{53}$$

The computation in (53) involves $s_{25,t}$ iterating on which, in turn, requires another off L-band element, $s_{15,t}$, and so on. Hence, a single iteration of the algorithm, although distributed and requiring only local communication, sweeps the entire rows in $\mathbf{S}$ and the computation requirements scale linearly with n. We now present a solution to this problem.

B. Distributed Iterate Collapse Inversion Overrelaxation Algorithm

In this section, we present the distribute iterate collapse inversion overrelaxation (DICI-OR) algorithm. The DICI-OR algorithm is divided into two steps: i) an iterate step and ii) a collapse step. The **iterate step** can be written, in general, for the ijth element that lies in the L-band $(|i-j| \leq L)$ of the matrix $\mathbf{S}_{t+1}$ as

$$s_{ij,t+1} = \begin{cases} \mathbf{p}_i\mathbf{s}_t^j, & i \neq j, \\ \mathbf{p}_i\mathbf{s}_t^j + \gamma m_{ii}^{-1}, & i = j, \end{cases} \quad |i-j| \leq L \tag{54}$$

where the symbols are defined as in Section V-A-2).

As we explained before in Section V-A-2), the implementation of (54) requires non-L-banded elements that, in turn, require more non-L-banded elements. To address this problem, we introduce a **collapse step**. We employ a determinant maximizing completion of $\mathbf{S}_t$ assuming that its non L-band elements are unspecified and use the results[9] in [37]. In general, a non-L-band element of an $L = 1$st-order Gauss-Markov covariance matrix (whose inverse is $L = 1$-banded) can be written as

$$s_{ij} = s_{i,j-1}s_{i+1,j-1}^{-1}s_{i+1,j}, \quad |i-j| > L, \tag{55}$$

which gives us the collapse step. In the context of Fig. 3, instead of iterating on s_{35} as in (53), we employ the collapse step

$$s_{35,t} = s_{34,t}s_{44,t}^{-1}s_{45,t} \tag{56}$$

that avoids the need to iterate further on the non-L-band elements.

Recall that at time k of the Information filter, we are interested in going from local information matrices $\mathbf{Z}_{k|k}^{(l)}$ to the local error covariances $\mathbf{S}_{k|k}^{(l)}$. Since at time k, the sensors have already computed the local error covariances at time $k-1$, we use them as the initial condition of the DICI algorithm. This choice of the initial conditions is an efficient starting point and provides a one-step convergence of the DICI algorithm in the steady state. The initial conditions of the DICI-OR algorithm are, thus, given by

$$\begin{aligned} \mathbf{P}_\gamma^{(l)} &= (1-\gamma)\mathbf{I}_{n\times n}^{(l)} \\ &\quad + \gamma\left(\mathbf{M}^{(l)}\right)^{-1}\left(\mathbf{M}^{(l)} - \mathbf{Z}_{k|k}^{(l)}\right) \end{aligned} \tag{57}$$

$$\mathbf{S}_{t=0,k|k}^{(l)} = \mathbf{S}_{k-1|k-1}^{(l)}. \tag{58}$$

Equations (57)–(58) do not require communication and can be computed at each sensor directly from the local information matrix, $\mathbf{Z}^{(l)}$. This is because the matrix $\mathbf{M}$ is diagonal, and $(\mathbf{M}^{(l)})^{-1}$ is the direct inverse of $\mathbf{M}^{(l)}$ in (57).

We refer to (54)–(55), combined with (57)–(58), and appropriate communication from neighboring sensors as the DICI-OR algorithm. The DICI-OR algorithm can be easily extended to $L > 1$. The only modification required is in the collapse step, since (55) holds only for $L = 1$. The appropriate formulas to replace (55), when $L > 1$, are provided in [37]. The computation requirements for the DICI-OR algorithm are independent of n, at each subsystem, and it provides a scalable implementation of the matrix inversion problem. The DICI algorithm (without the overrelaxation parameter, γ) can be obtained from the DICI-OR algorithm by setting $\gamma = 1$.

1) Convergence of the DICI-OR Algorithm: We introduce the following definition.

Definition 1: Let Ξ be the set of semiseparable matrices [48], [49] defined by

$$\Xi = \{\mathbf{S} \in \mathbb{R}^{n\times n} : \mathbf{S}^{-1} \text{ is an } L\text{-banded, symmetric positive definite matrix}\}. \tag{59}$$

The iterate and the collapse steps of the DICI algorithm can be combined in matrix form as follows:

$$\text{Iterate Step: } \mathbf{S}_{t+1} = \mathbf{P}_\gamma\overline{\mathbf{S}}_t + \gamma\mathbf{M}^{-1}, \quad |i-j| \leq L \tag{60}$$

$$\text{Collapse Step: } \overline{\mathbf{S}}_{t+1} = \zeta\left(\mathbf{S}_{t+1}\right), \quad |i-j| > L. \tag{61}$$

The operator $\zeta(\cdot)$ is the collapse operator; it employs a maximizing determinant/entropy completion of a covariance matrix whose non L-band elements are unspecified using the results in [37]. The DICI algorithm is a composition of the linear function (the iterate step in (60)), $\mathbf{D} : \overline{\mathbf{S}}_t \rightarrow \mathbf{P}_\gamma\overline{\mathbf{S}}_t + \gamma\mathbf{M}^{-1}$, followed by the collapse operator, $\zeta(\cdot)$ given in (55) for $L = 1$ and in

[9] If $\mathbf{S}$ is the inverse of an L-banded matrix, then the submatrices that do not lie in the L-band of $\mathbf{S}$, can be computed from the submatrices that lie in the L-band of $\mathbf{S}$, [37]. So, to compute the inverse $\mathbf{S} = \mathbf{Z}^{-1}$ of an L-banded matrix $\mathbf{Z}$, we just compute the submatrices that lie in the L-band of its inverse $\mathbf{S}$; the remaining submatrices are derived from these using the expressions given in [37].

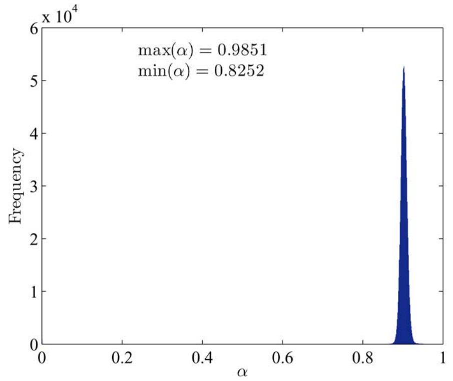


Fig. 5. Histogram of α: Simulations of the quotient of (64) are performed 1.003×10^6 times and the results are provided as a histogram.

[37] for $L > 1$. Combining (60)–(61) summarizes the DICI algorithm for $\overline{\mathbf{S}}_t \in \Xi$ as

$$\overline{\mathbf{S}}_{t+1} = \zeta\left(\mathbf{P}_\gamma \overline{\mathbf{S}}_t + \gamma \mathbf{M}^{-1}\right). \qquad (62)$$

We define a composition map, $\boldsymbol{\Upsilon} : \Xi \mapsto \Xi$, as $\boldsymbol{\Upsilon} \doteq \zeta \circ \mathbf{D}$. To prove the convergence of the DICI-OR algorithm, we show that the composition map, $\boldsymbol{\Upsilon}$, is a contraction map under some norm, that we choose to be the spectral norm $\|\cdot\|_2$, i.e., for some $\alpha \in [0, 1)$

$$\|\boldsymbol{\Upsilon}(\mathbf{X}_\Xi) - \boldsymbol{\Upsilon}(\mathbf{Y}_\Xi)\|_2 \leq \alpha \|\mathbf{X}_\Xi - \mathbf{Y}_\Xi\|_2 \quad \forall \mathbf{X}_\Xi, \mathbf{Y}_\Xi \in \Xi. \quad (63)$$

The convergence of the iterate step of the DICI-OR algorithm is based on the multiplier matrix $\mathbf{P}_\gamma$, which is proved to be a contraction map in [21]. For the convergence of the collapse operator ζ, we resort to a numerical procedure and show, in the following, that (63) is a contraction by simulating (63) 1.003×10^6 times.

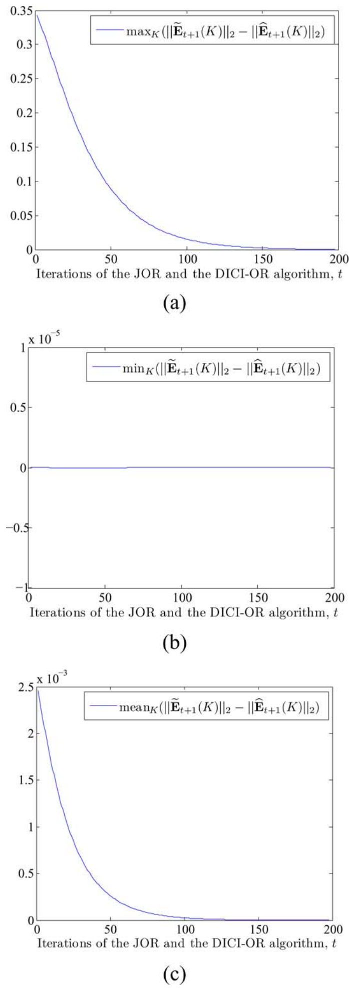
For the simulations, we generate $n \times n$ matrices $\mathbf{X}_{\text{rand}}$, with independent and identically distributed (i.i.d.) normally distributed elements and get $\mathbf{X}_{\text{sym}} = \mathbf{X}_{\text{rand}} + \mathbf{X}_{\text{rand}}^T$. We eigendecompose $\mathbf{X}_{\text{sym}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$. We replace $\boldsymbol{\Lambda}$ with a diagonal matrix $\boldsymbol{\Lambda}_\Xi$, whose diagonal elements are drawn from a uniform distribution in the interval $(0, 10]$. This leads to a random symmetric positive definite matrix that lies in $\mathcal{S}$ when we apply the collapse operator $\mathbf{X}_\Xi = \zeta\left(\mathbf{V}\boldsymbol{\Lambda}_\Xi\mathbf{V}^T\right)$. For $n = 100$ and L a random integer between 1 and $n/2 = 50$, we compute, by Monte Carlo simulations, the quotient of (63)

$$\frac{\|\boldsymbol{\Upsilon}(\mathbf{X}_\Xi) - \boldsymbol{\Upsilon}(\mathbf{Y}_\Xi)\|_2}{\|\mathbf{X}_\Xi - \mathbf{Y}_\Xi\|_2}. \qquad (64)$$

The number of trials is 1.003×10^6. The histogram of the values of α, in (64), (with 1000 bins) is plotted in Fig. 5. The maximum value of α found in these 1.003×10^6 simulations is 0.9851 and the minimum value is 0.8252. Since $\alpha \in (0, 1)$, i.e., strictly less than 1, we assume that (63) is numerically verified.

2) Error Bound for the DICI-OR Algorithm: Let the matrix produced by the DICI-OR algorithm at the $t + 1$th iteration be $\widehat{\mathbf{S}}_{t+1}$. The error process in the DICI-OR algorithm is given by

$$\widehat{\mathbf{E}}_{t+1} = \widehat{\mathbf{S}}_{t+1} - \mathbf{S}. \qquad (65)$$



(a)

(b)

(c)

Fig. 6. Simulation for the error bound of the DICI-OR algorithm.

Claim: The spectral norm of the error process $\|\widehat{\mathbf{E}}_{t+1}\|_2$ of the DICI-OR algorithm is *bounded above* by the spectral norm of the error process $\|\widetilde{\mathbf{E}}_{t+1}\|_2$ of the JOR algorithm. Since, the JOR algorithm always converges for symmetric positive definite matrices $\mathbf{Z}$, we deduce that the DICI-OR algorithm converges. We verify this claim numerically by Monte Carlo simulations. The number of trials is 4490, and we compute the error process $\widehat{\mathbf{E}}_{t+1}(K)$ of the DICI-OR algorithm and the error process $\widetilde{\mathbf{E}}_{t+1}(K)$ of the JOR algorithm. We choose the relaxation parameter γ to be 0.1. In Fig. 6(a)–(c), we show the following:

$$\{\max_K, \min_K, \ \text{mean}_K\}\left(\|\widetilde{\mathbf{E}}_{t+1}(K)\|_2 - \|\widehat{\mathbf{E}}_{t+1}(K)\|_2\right)$$

against the number of iterations of the JOR and DICI-OR algorithm. Since all the three figures show that the max, min, and the mean of the difference of the spectral norm of the two error processes $\|\widetilde{\mathbf{E}}_{t+1}(K)\|_2 - \|\widehat{\mathbf{E}}_{t+1}(K)\|_2$ is always ≥ 0, we deduce that [using (50)]

$$\|\widehat{\mathbf{E}}_{t+1}\|_2 \leq \|\widetilde{\mathbf{E}}_{t+1}\|_2 \leq \max_i |\lambda_{i(\mathbf{P}_\gamma^H \mathbf{P}_\gamma)}|^{t+1/2} \|\mathbf{S}_0 - \mathbf{S}\|_2.$$

This verifies our claim numerically and provides us an upper bound on the spectral norm of the error process of the DICI algorithm.

VI. Local Information Filters: Initial Conditions and Local Filter Step

The initial conditions and the local filter step of the LIFs are presented in Sections VI-A and B.

A. Initial Conditions

The initial condition on the local predictor is

$$\widehat{\mathbf{z}}^{(l)}_{0|-1} = \mathbf{0}. \tag{66}$$

Since the local information matrix and the local error covariances are not the inverse of each other, (43), we obtain the initial condition on the prediction information matrix by using the L-banded inversion theorem [17], provided in appendix A. This step may require a local communication step further elaborated in Section VII-A.

$$\mathbf{Z}^{(l)}_{0|-1} \xleftarrow{L\text{-Banded Inversion Theorem}} \mathbf{S}^{(l)}_{0}. \tag{67}$$

B. Local Filter Step

In this section, we present the local filter step of the LIFs. The local filter step is given by

$$\mathbf{Z}^{(l)}_{k|k} = \mathbf{Z}^{(l)}_{k|k-1} + \mathcal{I}^{(l)}_{f} \tag{68a}$$

$$\widehat{\mathbf{z}}^{(l)}_{k|k} = \widehat{\mathbf{z}}^{(l)}_{k|k-1} + \mathbf{i}^{(l)}_{f,k} \tag{68b}$$

where $\mathcal{I}^{(l)}_{f}$ and $\mathbf{i}^{(l)}_{f,k}$ denote the fused observation variables. Fusion of the observations is presented in Section IV-A. The distribution of the addition operation "+" in (22) is straightforward in (68). Recall that the observation fusion, (42), is carried out in a single-step for the sensor communication graph, $\mathcal{G}$, or using the iterative weighted averaging algorithm for the sensor communication graph $\widetilde{\mathcal{G}}$. In case of $\widetilde{\mathcal{G}}$, the asymptotic convergence of this iterative algorithm is guaranteed under certain conditions; see [13] for details on such conditions. Hence, with the required assumptions on the subgraph $\mathcal{G}_j$, the observation fusion algorithm, (42), asymptotically converges, and hence (with a slight abuse of notation),

$$\bigcup_{l=1}^{N} \mathbf{i}^{(l)}_{f,k} \rightarrow \mathbf{i}_k \text{ and } \bigcup_{l=1}^{N} \mathcal{I}^{(l)}_{f} \rightarrow \mathcal{I}. \tag{69}$$

The above notation implies that the local fused information variables, $\mathcal{I}^{(l)}_{f}$ and $\mathbf{i}^{(l)}_{f,k}$, when combined over the entire sensor network, asymptotically converge to the global information variables, $\mathcal{I}$ and $\mathbf{i}_k$. This, in turn, implies that the local filter step of the LIFs asymptotically converges to the global filter step, (22), of the CLBIF.

Once the local filter step is completed, the DICI algorithm is employed on the local information matrices $\mathbf{Z}^{(l)}_{k|k}$ obtained from (68a) and (68b) to convert them into the local error covariance matrices $\mathbf{S}^{(l)}_{k|k}$. Finally, to convert the estimates in the information domain $\widehat{\mathbf{z}}^{(l)}_{k|k}$ to the estimates in the Kalman filter domain $\widehat{\mathbf{x}}^{(l)}_{k|k}$, we specialize the DICI algorithm to the matrix-vector product (15) that does not require a collapse step; hence, convergence can be proved analytically.

VII. Local Information Filters: Local Prediction Step

This section presents the distribution of the global prediction step, (23), into the local prediction step at each LIF. This section requires the results of the DICI algorithm for the L-banded matrices, introduced in Section V.

A. Computing the Local Prediction Information Matrix $\mathbf{Z}^{(l)}_{k|K-1}$

Because of the coupled local dynamics of the reduced sensor-based models, each sensor may require that some of the estimated states be communicated as internal inputs $\widehat{\mathbf{d}}^{(l)}_{k|k}$ to its LIF, as shown in (37). These states are the directed edges into each cut-point set in Fig. 1(b). Hence, the error associated to a local estimation procedure is also influenced by the error associated to the neighboring estimation procedure, from where the internal inputs are being communicated. This dependence is true for all sensors and is reflected in the local prediction error covariance matrix $\mathbf{S}^{(l)}_{k|k-1}$, as it is a function of the global estimation error covariance matrix $\mathbf{S}_{k-1|k-1}$. Equation (70) follows from (23a) after expanding (23a) for each diagonal submatrix $\mathbf{S}^{(l)}_{k|k-1}$ in $\mathbf{S}_{k|k}$.

$$\mathbf{S}^{(l)}_{k|k-1} = \mathbf{F}_l \mathbf{S}_{k-1|k-1} \mathbf{F}_l^T + \mathbf{G}^{(l)} \mathbf{Q}^{(l)} \mathbf{G}^{(l)T}. \tag{70}$$

The matrix $\mathbf{F}_l = \mathbf{T}_l \mathbf{F}$ (the matrix $\mathbf{T}_l$ is introduced in (33)) is an $n_l \times n$ matrix, which relates the state vector $\mathbf{x}_k$ to the local state vector $\mathbf{x}^{(l)}_k$. Fig. 1(c) shows that the matrix $\mathbf{F}_l$ is further divided into $\mathbf{F}^{(l)}$ and $\mathbf{D}^{(l)}$. With this subdivision of $\mathbf{F}_l$, the first term on the right-hand side of (70), $\mathbf{F}_l \mathbf{S}_{k-1|k-1} \mathbf{F}_l^T$, can be expanded, and (70) can be written as

$$\begin{aligned}\mathbf{S}^{(l)}_{k|k-1} = {} & \mathbf{F}^{(l)} \mathbf{S}^{(l)}_{k-1|k-1} \mathbf{F}^{(l)T} + \mathbf{F}^{(l)} \mathbf{S}^{x^{(l)}d^{(l)}}_{k-1|k-1} \mathbf{D}^{(l)T} \\ & + \left(\mathbf{F}^{(l)} \mathbf{S}^{x^{(l)}d^{(l)}}_{k-1|k-1} \mathbf{D}^{(l)T}\right)^T \\ & + \mathbf{D}^{(l)} \mathbf{S}^{d^{(l)}d^{(l)}}_{k-1|k-1} \mathbf{D}^{(l)T} + \mathbf{G}^{(l)} \mathbf{Q}^{(l)} \mathbf{G}^{(l)T}\end{aligned} \tag{71}$$

where

$\mathbf{S}^{(l)}_{k-1\vert k-1}$	is the local error covariance matrix, which is available from (68a) and the DICI algorithm at sensor l;
$\mathbf{S}^{d^{(l)}d^{(l)}}_{k-1\vert k-1}$	is the local error covariance matrix, which is available from (68a) and the DICI algorithm at the sensors having the states, $\mathbf{d}^{(l)}_k$, in their reduced models;
$\mathbf{S}^{x^{(l)}d^{(l)}}_{k-1\vert k-1}$	is the error cross correlation between the local state vector, $\mathbf{x}^{(l)}_k$, and the local internal input vector, $\mathbf{d}^{(l)}_k$.

The non-L-banded entries in this matrix can be computed from (56); see [37]. Since the model matrix $\mathbf{F}$ is sparse, we do not need the entire error covariance matrix $\mathbf{S}_{k-1|k-1}$, only certain of its submatrices. Since the model matrix $\mathbf{F}$ is localized, long-distance communication is not required, and the submatrices are available at the neighboring sensors.

Once we have calculated the local prediction error covariance matrix $\mathbf{S}^{(l)}_{k|k-1}$, we realize (43) and compute the local prediction information matrix $\mathbf{Z}^{(l)}_{k|k-1}$, using the L-banded Inversion Theorem (see [17] and Appendix A).

$$\mathbf{Z}^{(l)}_{k|k-1} \xleftarrow[L\text{-Banded Inversion Theorem}]{} \mathbf{S}^{(l)}_{k|k-1}. \tag{72}$$

From (78) in Appendix A, to calculate the local prediction information matrix $\mathbf{Z}^{(l)}_{k|k-1}$, we only need the $\mathbf{S}^{(l)}_{k|k-1}$ from sensor "l" and from some additional neighboring sensors. Hence $\mathbf{Z}^{(l)}_{k|k-1}$ is again computed with only local communication and n_lth order computation.

B. Computing the Local Predictor, $\widehat{\mathbf{z}}^{(l)}_{k|k-1}$

We illustrate the computation of the local predictor $\widehat{\mathbf{z}}^{(3)}_{k|k-1}$ for the 5-D system, (26)–(27), with $L=1$. The local predictor $\widehat{\mathbf{z}}^{(3)}_{k|k-1}$ at sensor 3 follows from the global predictor, (23b), and is given by

$$\begin{aligned}\widehat{\mathbf{z}}^{(3)}_{k|k-1} = {} & \mathbf{Z}^{(3)}_{k|k-1}\left(\mathbf{F}^{(3)}\widehat{\mathbf{x}}^{(3)}_{k-1|k-1} + \mathbf{D}^{(3)}\widehat{\mathbf{d}}^{(3)}_{k-1|k-1}\right) \\ & + \begin{bmatrix} z_{34}\left(f_{31}\widehat{x}^{(1)}_{1,k-1|k-1} + f_{33}\widehat{x}^{(2)}_{3,k-1|k-1}\right) \\ 0 \end{bmatrix}\end{aligned} \tag{73}$$

where z_{34} is the only term arising due to the $L=1$-banded (tridiagonal) assumption on the prediction information matrix $\mathbf{Z}_{k|k-1}$. Note that $f_{31}\widehat{x}^{(1)}_{1,k-1|k-1} + f_{33}\widehat{x}^{(2)}_{3,k-1|k-1}$ is a result of $\mathbf{f}_3\widehat{\mathbf{x}}_{k-1|k-1}$, where $\mathbf{f}_3$ is the third row of the model matrix $\mathbf{F}$. A model matrix with a localized and sparse structure ensures that $\mathbf{f}_3\widehat{\mathbf{x}}_{k-1|k-1}$ is computed from a small subset of the estimated state vector $\widehat{\mathbf{x}}^{(\mathcal{Q})}_{k-1|k-1}$, communicated by a subset $\mathcal{Q} \subseteq \mathcal{K}(l)$ of the neighboring sensors, which are modeling these states in their reduced models. This may require multihop communication.

Generalizing, the local predictor in the information domain $\widehat{\mathbf{z}}^{(l)}_{k|k-1}$ is given by

$$\begin{aligned}\widehat{\mathbf{z}}^{(l)}_{k|k-1} = {} & \mathbf{Z}^{(l)}_{k|k-1}\left(\mathbf{F}^{(l)}\widehat{\mathbf{x}}^{(l)}_{k-1|k-1} + \mathbf{D}^{(l)}\widehat{\mathbf{d}}^{(l)}_{k-1|k-1}\right) \\ & + f_1\left(\mathbf{Z}^{(\mathcal{V})}_{k|k-1}, \mathbf{F}^{(\mathcal{V})}, \widehat{\mathbf{x}}^{(\mathcal{Q})}_{k-1|k-1}\right)\end{aligned} \tag{74}$$

for some $\mathcal{V}, \mathcal{Q} \subseteq \mathcal{K}(l)$, where $f_1(\cdot)$ is a linear function and depends on L.

C. Estimate Fusion

We present the following fact, if any $\mathcal{G}_j$ is not fully connected.

1) Fact: Let m denote the number of iterations of the consensus algorithm that are employed to fuse the observations [recall (42)]. As $m \to \infty$, the local estimates, $\widehat{\mathbf{z}}^{(l)}_{k|k}$, in (68b) also reach a consensus on the estimates of the shared states, i.e., the local estimates converge to the CLBIF estimates.

It is straightforward to note that, *if* the local predictors $\widehat{\mathbf{z}}^{(l)}_{k|k-1}$ of the shared states are the same over the sensors that share these states, *then* as $m \to \infty$ we have a consensus on the estimates (of the shared states) in the local filter step (68b). To show that the shared local predictors are the same over the sensors that share these states, we refer back to our illustration and write the local predictors for sensor 2 as follows:

$$\begin{aligned}\widehat{\mathbf{z}}^{(2)}_{k|k-1} = {} & \mathbf{Z}^{(2)}_{k|k-1}\left(\mathbf{F}^{(2)}\widehat{\mathbf{x}}^{(2)}_{k-1|k-1} + \mathbf{D}^{(2)}\widehat{\mathbf{d}}^{(2)}_{k-1|k-1}\right) \\ & + \begin{bmatrix} z_{12}\left(f_{11}\widehat{x}^{(1)}_{1,k-1|k-1} + f_{12}\widehat{x}^{(2)}_{2,k-1|k-1}\right) \\ 0 \\ z_{45}\left(f_{54}\widehat{x}^{(2)}_{4,k-1|k-1} + f_{55}\widehat{x}^{(3)}_{5,k-1|k-1}\right) \end{bmatrix}.\end{aligned} \tag{75}$$

The predictor for the shared state $x_{4,k}$ can now be extracted for sensor 3 from (73) and for sensor 2 from (75) and can be verified to be the following for each $l = 2, 3$.

$$\begin{aligned}\widehat{z}^{(l)}_{4,k|k-1} = {} & z_{34}f_{31}\widehat{x}^{(1)}_{1,k-1|k-1} \\ & + (z_{34}f_{33} + z_{44}f_{43})\widehat{x}^{(2)}_{3,k-1|k-1} \\ & + z_{45}f_{54}\widehat{x}^{(3)}_{4,k-1|k-1} \\ & + (z_{44}f_{45} + z_{45}f_{55})\widehat{x}^{(3)}_{5,k-1|k-1}.\end{aligned} \tag{76}$$

The elements z_{ij} belong to the prediction information matrix, which is computed using the DICI algorithm and the L-banded inversion theorem. It is noteworthy that the DICI algorithm is not a consensus algorithm and thus the elements z_{ij} are the same across the sensor network at any iteration of the DICI algorithm. With the same local predictors, the iterations of the consensus algorithm on the observations lead to a consensus on the shared estimates.

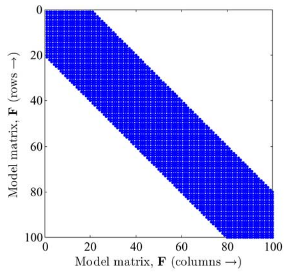
VIII. Results

A. Summary of the Local Information Filters

We summarize the distributed local Information filters. The initial conditions are given by (66) and (67). Observation fusion is carried out using (42). The fused observation variables $\mathbf{i}^{(l)}_{f,k}$ and $\mathbf{I}^{(l)}_{f,k}$ are then employed in the local filter step, (68a) and (68b), to obtain the local information matrix and the local estimator $\widehat{\mathbf{Z}}^{(l)}_{k|k}$ and $\mathbf{z}^{(l)}_{k|k}$, respectively. We then implement the DICI algorithm (54) and (55) to compute the local error covariance matrix $\mathbf{S}^{(l)}_{k|k}$ from the local information matrix $\mathbf{Z}^{(l)}_{k|k}$. The DICI algorithm is again employed to compute the local estimates in the Kalman filter domain $\widehat{\mathbf{x}}^{(l)}_{k|k}$ from the local estimator $\widehat{\mathbf{z}}^{(l)}_{k|k}$ as a special case. Finally, the local prediction step is completed by computing the local prediction error covariance matrix $\widehat{\mathbf{S}}^{(l)}_{k|k-1}$, the local prediction information matrix $\widehat{\mathbf{Z}}^{(l)}_{k|k-1}$, and the local predictor $\widehat{\mathbf{z}}^{(l)}_{k|k-1}$ from (71), (72), and (74), respectively.

B. Simulations

We simulate an $n = 100$-dimensional system with $N = 10$ sensors monitoring the system. Fig. 7(a) and (b) shows the nonzero elements (chosen at random) of the model matrix, $\mathbf{F}$, such that its maximum eigenvalue is unity, i.e., $\max_{i\lambda_i(\mathbf{F})} = 1$. The model matrix in Fig. 7(a) is $L = 20$-banded. The model matrix in Fig. 7(b) is $L = 36$-banded that is obtained by employing



(a) (b)

Fig. 7. (a), (b) Nonzero elements (chosen at random) of 100×100, $L = 20$-banded [Fig. 7(a)] and $L = 36$-banded [Fig. 7(b)] model matrices, $\mathbf{F}$, such that $\|\mathbf{F}\|_2 = 1$.

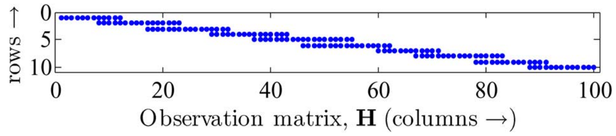
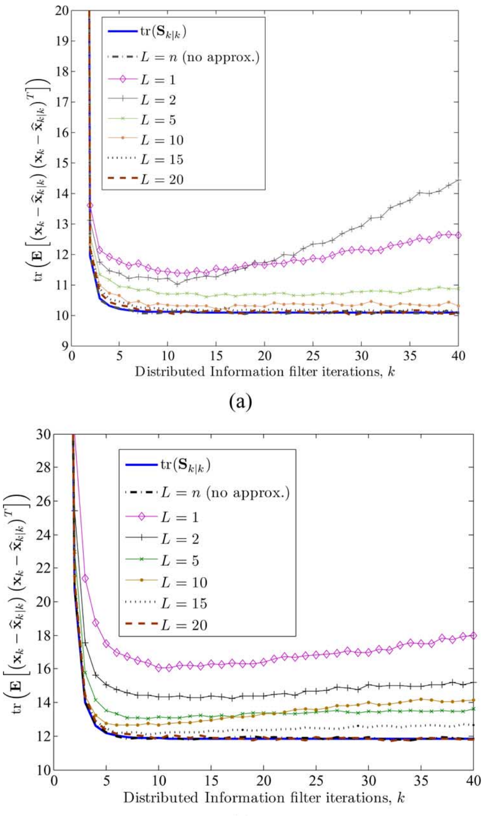


Fig. 8. Global observation matrix, $\mathbf{H}$. The nonzero elements (chosen at random) are shown. There are $N = 10$ sensors, where the lth row of $\mathbf{H}$ corresponds to the local observation matrix, $\mathbf{H}_l$, at sensor l. The overlapping states (for which fusion is required) can be seen as the overlapping portion of the rows.

the reverse Cuthill–Mckee algorithm [34] for bandwidth reduction of a sparse random $\mathbf{F}$. The nonzeros (chosen at random as Normal(0,1)) of the global observation matrix $\mathbf{H}$ are shown in Fig. 8. The lth row of the global 10×100 observation matrix $\mathbf{H}$ is the local observation matrix $\mathbf{H}_l$ at sensor l. Distributed Kalman filters are implemented on i) $\mathbf{F}$ in 7(a) and $\mathbf{H}$ in Fig. 8; and ii) $\mathbf{F}$ in 7(b) and $\mathbf{H}$ in Fig. 8. The trace of the error covariance matrix $\mathbf{S}_{k|k}$ is simulated for different values of L in $[n, 1, 2, 5, 10, 15, 20]$ and the plots are shown (after averaging over 1000 Monte Carlo trials) in Fig. 9(a) for case i) and in Fig. 9(b) for case ii). The stopping criteria for the DICI algorithm and the consensus algorithm are such that the deviation in their last 10 iterations is less than 10^{-5}. In both Fig. 9(a) and (b), $\mathrm{tr}(S_{k|k})$ represents the trace of the solution of the Riccati equation in the CIF (no approximation).

With 1000 Monte Carlo trials, we further simulate the trace of the error covariance, $\mathrm{tr}(S_{k|k})$, for case ii) with $L = 20$-banded approximation (on the information matrices) as a function of the number of iterations, t, of the DICI-OR algorithm. We compare this with a) the simulation obtained from the $O(n^3)$ direct inverse of the error covariance (with $L = 20$-banded approximation on its inverse) and b) $\mathrm{tr}\left(\mathbf{S}_{k|k}\right)$, trace of the solution of the Riccati equation of the CIF (no approximation). We choose $t = [1,10,30,100,200]$ for the DICI algorithm and show the results in Fig. 10. As $t \uparrow$, the curves we obtain from the DICI algorithm get closer to the curve we obtain with the direct inverse.

The simulations confirm the following. First, the LIFs asymptotically track the results of the CLBIF (see Fig. 10). Second, we verify that as $L \uparrow$, the performance is virtually indistinguishable from that of the CIF, as pointed out in [37]; this is in agreement with the fact that the approximation is optimal in Kullback–Leibler sense, as shown in [17]. Here, we also point out that, as we increase L, the performance increases, but, we pay a price in terms of the communication cost, as we may have to communicate in a larger neighborhood. Third, in Fig. 9(a) and (b), it can be verified that L should be greater than some $L_{\min}$ for the CLBIF to converge, as pointed out in Section II-C. Fourth, since the trace of the error covariance is the trace of an expectation operator, we use Monte Carlo trials to simulate the expectation operator. If we increase the number of Monte Carlo trials the variations reduce, and the filters eventually follow the solution of the Riccati equation $\mathrm{tr}\left(\mathbf{S}_{k|k}\right)$. Fifth, the curve in Fig. 10 with $t = 1$ shows the decoupled LIFs, when the local error covariances are not assimilated. This investigates the case where the distributed estimation scheme



(b)

Fig. 9. (a), (b) Distributed Kalman filter is implemented on the model matrices in Fig. 7(a) and (b) and the global observation matrix $\mathbf{H}$ (Fig. 8), in Fig. 9(a) and (b). The expectation operator in the trace (on horizontal axis) is simulated over 1000 Monte Carlo trials.

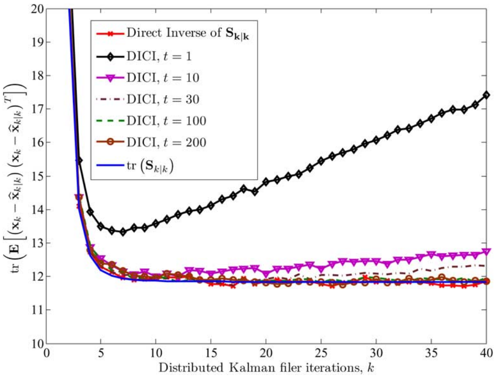


Fig. 10. Performance of the DICI algorithm as a function of the number of DICI iterations, t.

$$\mathbf{Z} = \begin{bmatrix} \boxed{\mathbf{S}^{1}_{L+1}}^{-1} - \boxed{\mathbf{S}^{2}_{L+1}}^{-1} + \boxed{\mathbf{S}^{2}_{L+2}}^{-1} - & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & + \boxed{\mathbf{S}^{N-L-1}_{N-1}}^{-1} - \boxed{\mathbf{S}^{N-L}_{N-1}}^{-1} + \boxed{\mathbf{S}^{N-L}_{N}}^{-1} \end{bmatrix} \tag{78}$$

fails because the covariances at the local filters are decoupled. We next discuss the computational advantage of the LIFs over some of the existing methods.

C. Complexity

We regard the multiplication of two $n \times n$ matrices as an $O(n^3)$ operation, inversion of an $n \times n$ matrix also as an $O(n^3)$ operation, and multiplication of an $n \times n$ matrix and an $n \times 1$ vector as an $O(n^2)$ operation. For all of the following, we assume N sensors monitoring the global system, (6).

1) Centralized Information Filter: This is the case where each sensor sends its local observation vector to a centralized location or a fusion center, where the global observation vector is put together. The fusion center implements the CIF, with an $O(n^3)$ computation complexity for each time k, so we have the complexity as $O(n^3 k)$, with inordinate communication requirements (back and forth communication between the sensors and the central location).

2) Information Filters With Replicated Models at Each Sensor and Observation Fusion: In this scheme, the local observation vectors are not communicated to the central location, but are fused over an all-to-all communication network [3] or an arbitrary network [7]. Computational complexity at each sensor is $O(n^3 k)$ in [3]. In [7], let the complexity of the iterative algorithm be $O(\varphi(n))$ and let t_φ be the number of iterations of the consensus algorithm. Each sensor implements a CIF after fusing the observations. For each time index k, each sensor requires $O(n^3)$ operations plus the operations required for the consensus algorithm, which are $O(\varphi(n)t_\varphi)$; so the total computation complexity is $O\left(\left(n^3 + \varphi(n)t_\varphi\right)k\right)$ at each sensor. Communication requirements are global in [3] and local in [7].

3) Distributed Kalman Filters: Local Information Filters: The distributed Kalman filter presented in this paper has three iterative algorithms. In all other steps, the computation complexity is dominated by $O\left(n_l^3\right)$, where $n_l \ll n$. Let t_o be the iterations required by the weighted averaging algorithm, where at each step of the iterative algorithm the computations are dominated by $O\left(n_l^2\right)$. Let t_{J_1} be the iterations required by the DICI algorithm for vectors, where at each step of the iterative algorithm the computations are dominated by an $O(L^2)$ operations. Let t_{J_2} be the iterations required by the DICI algorithm, where at each step of the iterative algorithm the computations are dominated by $O(L^4)$ operations. Recalling that $L \sim n_l$ from Section III, the total computation complexity is $O\left(\left(n_l^3 + n_l^2 t_o + n_l^2 t_{J_1} + n_l^4 t_{J_2}\right)k\right)$. Let $t_\varpi = \max(t_o, t_{J_1}, t_{J_2})$, then the computation complexity is bounded by $O\left(n_l^4 t_\varpi k\right)$ at each sensor for the LIFs, which is much smaller than the computational cost of the solutions in VIII-C-1) and VIII-C-2). The communication requirement in the LIFs may be multihop but is always constrained to a neighborhood because of the structural assumptions on the model matrix $\mathbf{F}$.

IX. Conclusion

In conclusion, we presented a *distributed* implementation of the Kalman filter for sparse large-scale systems monitored by sensor networks. In our solution, the communication, computing, and storage is local and distributed across the sensor network, no single sensor processes n-dimensional vectors or matrices, where n, usually a large number, is the dimension of the state vector representing the random field. We achieve this by solving three linked problems: 1) *Spatial decomposition* of the global dynamical system into subsystems; these subsystems are obtained using a graph-theoretic model distribution technique; 2) *fusing*, through distributed averaging, multiple sensor observations of the state variables that are common across subsystems; and 3) *inverting* full n-dimensional matrices, with local communication only, by deriving an iterative DICI algorithm. The DICI algorithm only requires matrices and vectors of order n_l of the reduced state vectors. The DICI algorithm preserves the coupling among the local Kalman filters. Our solution is optimal when the error processes of the Kalman filter are constrained to Gauss–Markov random processes and contrasts with existing Kalman filter solutions for sensor networks that either replicate an n-dimensional Kalman filter at each sensor or reduce the model dimension at the expense of decoupling the field dynamics into lower-dimensional models. The former are infeasible for large-scale systems and the latter are not optimal and further cannot guarantee any structure of the centralized error covariances.

Simulations show that the distributed implementation with local Kalman filters implemented at each sensor converges to the global Kalman filter as the bandwidth L of the approximated information matrices is increased.

Appendix
L-Banded Inversion Theorem

Let $\mathbf{Z} = \mathbf{S}^{-1}$ be L-banded. We apply the algorithm, given in [17], to obtain $\mathbf{Z}$ from the submatrices in the L-band of $\mathbf{S}$. We use the following notation, in (77), to partition matrix addition and subtraction in terms of its constituent submatrices. Also $\mathbf{S}^i_j$

represents the principal submatrix of $\mathbf{S}$ spanning rows i through j, and columns i through j.

$$\begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & x+y+z & a_4 \\ 0 & a_5 & a_6 \end{bmatrix} = \left[\boxed{\begin{matrix} a_1 & a_2 \\ a_3 & x \end{matrix}} + \boxed{y} + \boxed{\begin{matrix} z & a_4 \\ a_5 & a_5 \end{matrix}} \right] \tag{77}$$

The inverse of $\mathbf{S}$, when $\mathbf{Z} = \mathbf{S}^{-1}$ is L-banded, is given by (78) shown at the top of the previous page, taken from [17], in terms of the L-banded submatrices of $\mathbf{S}$. Note that, to compute a principal submatrix in $\mathbf{Z}$, we do not need the entire $\mathbf{S}$, or even all the L-banded submatrices in $\mathbf{S}$. Instead, we only require neighboring submatrices in the L-band of $\mathbf{S}$. For proofs and further details, the interested reader can refer to [17].

References

[1] R. Kalman, “A new approach to linear filtering and prediction problems,” *Trans. ASME—J. Basic Eng.*, vol. 82, no. 2, pp. 35–45, 1960.

[2] R. Kalman and R. Bucy, “New results in linear filtering and prediction theory,” *ASME J. Basic Eng.*, vol. 83, pp. 95–108, 1961.

[3] B. Rao and H. Durrant-Whyte, “Fully decentralized algorithm for multisensor Kalman filtering,” *Proc. Inst. Electr. Eng.—Control Theory Appl.*, vol. 138, pp. 413–420, Sep. 1991.

[4] V. Saligrama and D. Castanon, “Reliable distributed estimation with intermittent communications,” in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, Dec. 2006, pp. 6763–6768.

[5] T. Chung, V. Gupta, J. Burdick, and R. Murray, “On a decentralized active sensing strategy using mobile sensor platforms in a network,” in *Proc. 43rd IEEE Conf. Decision Control*, Paradise Island, Bahamas, Dec. 2004, vol. 2, pp. 1914–1919.

[6] H. Hashemipour, S. Roy, and A. Laub, “Decentralized structures for parallel Kalman filtering,” *IEEE Trans. Autom. Control*, vol. 33, no. 1, pp. 88–94, Jan. 1988.

[7] R. Olfati-Saber, “Distributed Kalman filters with embedded consensus filters,” in *Proc. 44th IEEE Conf. Decision Control*, Seville, Spain, Dec. 2005, pp. 8179–8184.

[8] R. Olfati-Saber and J. Shamma, “Consensus filters for sensor networks and distributed sensor fusion,” in *Proc. 44th IEEE Conf. Decision Control*, Seville, Spain, Dec. 2005, pp. 6698–6703.

[9] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, “Kalman filter with intermittent observations,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[10] T. Berg and H. Durrant-Whyte, “Model distribution in decentralized multi-sensor data fusion,” Univ. of Oxford, Oxford, U.K., Tech. Rep., 1990.

[11] A. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton, FL: CRC Press, 1998.

[12] U. A. Kahn and J. M. F. Moura, “Distributed Kalman filters in sensor networks: Bipartite fusion graphs,” in *Proc. 15th IEEE Workshop Statistical Signal Processing*, Madison, WI, Aug. 26–29, 2007, pp. 700–704.

[13] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Syst. Controls Lett.*, vol. 53, no. 1, pp. 65–78, Apr. 2004.

[14] D. D. Šiljak, *Decentralized Control of Complex Systems*. Boston, MA: Academic, 1991.

[15] R. Grone, C. Johnson, E. Sa, and H. Wolkowicz, “Positive definite completions of partial Hermitian matrices,” *Linear Algebra Its App.*, vol. 58, pp. 109–124, Apr. 1984.

[16] N. Balram and J. M. F. Moura, “Noncausal Gauss Markov random fields: Parameter structure and estimation,” *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1333–1355, Jul. 1993.

[17] A. Kavcic and J. M. F. Moura, “Matrices with banded inverses: Inversion algorithms and factorization of Gauss–Markov processes,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1495–1509, Jul. 2000.

[18] A. B. Frakt, H. Lev-Ari, and A. S. Willsky, “A generalized Levinson algorithm for covariance extension with application to multiscale autoregressive modeling,” *IEEE Trans. Inf. Theory*, vol. 49, pp. 411–424, Feb. 2003.

[19] U. A. Khan and J. M. F. Moura, “Distributed iterate-collapse inversion (DICI) algorithm for L-banded matrices,” in *Proc. 33rd Int. Conf. Acoustics, Speech, Signal Processing*, Las Vegas, NV, Mar. 04, 2008, pp. 2529–2532.

[20] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[22] W. Ledsham and D. Staelin, “An extended Kalman–Bucy filter for atmospheric temperature profile retrieval with a passive microwave sounder,” *J. Appl. Meteorol.*, vol. 17, pp. 1023–1033, Jul. 1978.

[23] R. Brammer, “Estimation of the ocean geoid near the Blake escarpment using geos-3 satellite altimetry data,” *J. Geophys. Res.*, vol. 84, no. B8, pp. 3843–3852, Jul. 1979.

[24] J. Galantowicz, D. Entekhabi, and E. Njoku, “Tests of sequential data assimilation for retrieving profile soil moisture and temperature from observed L-band radiobrightness,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 4, pp. 1860–1870, Jul. 1999.

[25] M. Buehner and P. Malanotte-Rizzoli, “Reduced-rank Kalman filters applied to an idealized model of the wind-driven ocean circulation,” *J. Geophys. Res.*, vol. 108, no. C6, pp. 3192–3192, 2003.

[26] M. Buehner, P. Malanotte-Rizzoli, A. Busalacchi, and T. Inui, “Estimation of the tropical Atlantic circulation from altimetry data using a reduced-rank stationary Kalman filter,” *Interhemispheric Water Exchanges Atlantic Ocean* , ser. Elsevier Oceanographic Series, vol. 68, no. 9, pp. 49–92, 2003.

[27] A. Graham, *Kronecker Products and Matrix Calculus With Applications*. Chichester, U.K.: Ellis Horwood, 1981.

[28] N. Motee and A. Jadbabaie, “Optimal control of spatially distributed systems,” in *Proc. Amer. Control Conf.*, New York, Jul. 2007, pp. 778–783.

[29] J. Brailean and A. Katsaggelos, “Simultaneous recursive displacement estimation and restoration of noisy-blurred image sequences,” *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1236–1251, Sep. 1995.

[30] F. Khellah, P. Fieguth, M. Murray, and M. Allen, “Statistical processing of large image sequences,” *IEEE Trans. Image Process.*, vol. 14, no. 1, pp. 80–93, Jan. 2005.

[31] A. Bergen and D. Hill, “A structure preserving model for power system stability analysis,” *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 1, pp. 25–35, Jan. 1981.

[32] M. Ilic, E. Allen, J. Chapman, C. King, J. Lang, and E. Litvinov, “Preventing future blackouts by means of enhanced electric power systems control: From complexity to order,” *Proc. IEEE*, vol. 93, pp. 1920–1941, Nov. 2005.

[33] M. D. Ilić, L. Xie, U. A. Khan, and J. M. F. Moura, “Modelling future cyber-physical energy systems,” in *Proc. IEEE Power Eng. Soc. General Meeting*, Pittsburgh, PA, Jul. 2008.

[34] E. Cuthill and J. McKee, “Reducing the bandwidth of sparse symmetric matrices,” in *Proc. 24th Nat. Conf.*, New York, 1969, pp. 157–172.

[35] T. Chin, W. Karl, and A. Willsky, “Sequential filtering for multi-frame visual reconstruction,” *Special Issue on Multidimensional Signal Processing*, vol. 28, no. 3, pp. 311–333, 1992.

[36] W. W. Barrett and P. J. Feinsilver, “Gaussian families and a theorem on patterned matrices,” *J. Appl. Probab.*, vol. 15, no. 3, pp. 514–522, Sep. 1978.

[37] A. Asif and J. M. F. Moura, “Inversion of block matrices with L-block banded inverse,” *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 630–642, Feb. 2005.

[38] H. Zhang, J. M. F. Moura, and B. Krogh, “Estimation in sensor networks: A graph approach,” in *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 203–209.

[39] B. Béla, *Modern Graph Theory*. New York: Springer, 1998.

[40] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed Kalman filtering based on consensus strategies,” Information Engineering Dept., Univ. of Padova, Padova, Italy, Tech. Rep., 2007.

[41] L. Xiao and S. Boyd, “Designing fast distributed iterations via semidefinite programming,” presented at the Workshop on Large Scale Nonlinear and Semidefinite Programming, Waterloo, Canada, May 2004.

[42] S. Kar and J. M. F. Moura, “Sensor networks with random links: Topology design for distributed consensus,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pt. 2, Jul. 2008.

[43] J. Garloff, “Block methods for the solution of linear interval equations,” *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 1, pp. 89–106, Jan. 1990.

[44] A. I. Zečević and D. D. Šiljak, “A block-parallel Newton method via overlapping epsilon decompositions,” *SIAM J. Matrix Anal. Appl.*, vol. 15, no. 3, pp. 824–844, 1994.

[45] D. D. Šiljak and A. I. Zečević, “Overlapping block-iterative methods for solving algebraic equations,” *J. Difference Equations Appl.*, vol. 1, pp. 125–136, 1995.

[46] D. M. Stipanovic and D. D. Šiljak, “Jacobi and Gauss-Seidel iterations for polytopic systems: Convergence via convex M-matrices,” *Reliable Comput.*, vol. 6, no. 2, pp. 123–137, May 2000.

[47] B. Baran, E. Kaszkurewicz, and A. Bhaya, "Parallel asynchronous team algorithms: convergence and performance analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 7, pp. 677–688, Jul. 1996.

[48] F. R. Gantmacher and M. G. Krein, "Sur les matrices complètement non négatives et oscillatoires," *Compositio Mathematica*, vol. 4, pp. 445–476, 1937.

[49] R. Vandebril, M. V. Barel, G. Golub, and N. Mastronardi, "A bibliography on semiseparable matrices," *Calcolo*, vol. 43, no. 3–4, pp. 249–270, 2005.

Usman A. Khan (S'99) received the B.S. degree (with honors) in electrical engineering from the University of Engineering and Technology, Lahore-Pakistan, in 2002, and the M.S. degree in electrical and computer engineering from the University of Wisconsin-Madison in 2004. Currently, he is working towards the Ph.D. degree in electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA, under the supervision of Prof. José M. F. Moura.

Mr. Khan worked as a researcher in the National Magnetic Resonance Facility at Madison (NMRFAM), Madison, WI, from 2003 to 2005, where he worked on shape characterization of protein molecules. From 2004 to 2005, he was a Research Assistant in the Computer Science Department at the University of Wisconsin-Madison, where he worked on statistical estimation techniques on NMR signals. He worked as an intern in AKAMAI Technologies in summer 2007, where he implemented a control system for Internet load balancing. His research interests include statistical signal processing for large-scale dynamical systems, networked control and estimation, and distributed linear/nonlinear iterative algorithms.

Mr. Khan holds the ECE fellowship from the Carnegie Mellon University.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the Engenheiro Electrotécnico degree from the Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge.

He is a Professor of electrical and computer engineering and, by courtesy, of biomedical engineering, at Carnegie Mellon University (CMU), Pittsburgh, PA. He was on the faculty at IST, has held visiting faculty appointments at MIT, and was a visiting research scholar at the University of Southern California. He is a Founding Co-Director of the Center for Sensed Critical Infrastructures Research (CenSCIR) and manages a large education and research program between CMU and Portugal (www.icti.cmu.edu). His research interests include statistical and algebraic signal processing, image, bioimaging, and video processing, and digital communications. He has published over 300 technical journal and conference papers, is the co-editor of two books, holds six patents, and has given numerous invited seminars at international conferences, U.S. and European universities, and industrial and government Laboratories.

Dr. Moura is the President (2008–2009) of the IEEE Signal Processing Society (SPS). He was Editor-in-Chief for the IEEE TRANSACTIONS IN SIGNAL PROCESSING, interim Editor-in-Chief for the IEEE SIGNAL PROCESSING LETTERS, and was on the Editorial Board of several journals, including the IEEE PROCEEDINGS, the *IEEE Signal Processing Magazine*, and the ACM *Transactions on Sensor Networks*. He was on the steering and technical committees of several Conferences. He is a Fellow of the American Association for the Advancement of Science (AAAS), and a corresponding member of the Academy of Sciences of Portugal (Section of Sciences). He was awarded the 2003 IEEE Signal Processing Society Meritorious Service Award and in 2000 the IEEE Millennium Medal. In 2007, he received the CMU's College of Engineering Outstanding Research Award. He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.