Unifying Consensus and Covariance Intersection for Efficient Distributed State Estimation over Unreliable Networks

Amirhossein Tamjidi*, Reza Oftadeh†, Mohamed Naveed Gul Mohamed*, Dan Yu‡, Suman Chakravorty*, and Dylan Shell†
*Department of Aerospace Engineering, Texas A&M University
†Department of Computer Science & Engineering, Texas A&M University
‡College of Astronautics, Nanjing University of Aeronautics and Astronautics

***Abstract*—This paper presents and studies a recursive information consensus filter for decentralized dynamic-state estimation under circumstances in which the communication network is unreliable. Local estimators are assumed to have access only to local information and no structure is assumed about the topology of the communication network, which need not be connected at all times. The filter is a hybrid approach: it uses Iterative Covariance Intersection (ICI) to reach consensus over priors which might become correlated, while consensus over new information is handled using weights based on a Metropolis Hastings Markov Chain (MHMC). We establish bounds for estimation performance and show that this Hybrid method produces unbiased conservative estimates that are better than CI. The performance of the Hybrid method is evaluated extensively, including comparisons with competing algorithms, with a hypothetical 'full history' yardstick, and centralized performance. We conduct an assessment on a realistic atmospheric dispersion problem, and also on more carefully crafted settings to help characterize particular aspects of the performance.**

***Index Terms*—Distributed State Estimation, Covariance Intersection, Consensus Estimation.**

I. Introduction

Estimation as a way of fusing information from multiple sources connected via a network has many applications and, thus, has been extensively studied in recent years [1], [2]. In a sensor network, nodes represent sensors that make noisy observations of the state of an underlying system of interest. The estimation process is considered centralized if all the nodes send their raw observations to a central node which then calculates an estimate based on the collective information [3]. This is not always possible due to link failures as well as bandwidth and energy constraints [4]. One viable alternative, explored in this paper, is Distributed State Estimation (DSE).

In DSE, the processor on each node fuses local information with the incoming information from neighboring nodes and redistributes the fused result on the network. The objective is to design both a protocol for message passing between nodes and local fusion rules so that the nodes reach a consensus over their collective information. Although DSE algorithms are not guaranteed to match the performance of the centralized estimator all the time, their scalability, modularity, and robustness to network failure motivates the ongoing research. These features are important for the envisioned applications of such algorithms like multi-agent localization [5] and cooperative target tracking [6].

Manuscript received December 1, 2012; revised August 26, 2015. Corresponding author: Reza Oftadeh (email: reza.oftadeh@tamu.edu).

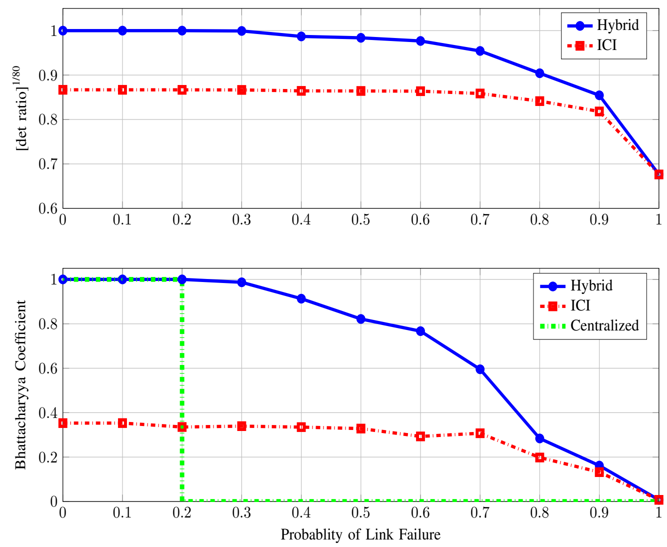


Fig. 1: Comparison of Iterative Covariance Intersection (ICI) and the Hybrid method studied in this paper. The graphs show average performance under different probabilities of link failure. Note that methods in [7], [8], [9] are only applicable in the area where the probability of link failure is less than 0.2. The Hybrid method relaxes the connectivity requirement and continues to achieve a better performance than ICI under network failure conditions.

DSE algorithms can be categorized based on the assumptions they make. Any DSE method makes assumptions about one or more of the following aspects: the state (static [10] or dynamic [5]), state transition model (linear [11] or non-linear [7]), type of noise (Gaussian [10], [11] or non-Gaussian [12]), topology of the network (constant or changing [13], [10]), connectivity of the network (always [7] or intermittent connection [13], [10]), agents' knowledge about the network topology (global or local [13], [10], [7]) and finally the treatment of mutual information between local estimate (exact solution through bookkeeping [1] or conservative solutions that avoid double counting [14], [15], [16]).

The research on DSE for linear systems with Gaussian noise is extensive (see [11], [17] for reviews). For nonlinear systems with Gausssian noise, the distributed versions of Extended Kalman Filters (EKF), Extended Information Filters (EIF), Unscented Kalman Filter (UKF), and Unscented Information Filter (UIF) have been proposed by [9], [18], [7], [19], respectively. For nonlinear systems with non-Gaussian noise, different flavors of Distributed Particle Filter (DPF) methods were proposed by [20]. In order to avoid scalability problems and the need for synchronized random generators, DPF methods make approximations that result in loss of performance compared to a centralized PF [12].

For dynamic systems, the connectivity constraint is a determining factor for choosing the proper DSE method. If the network remains connected, DSE methods can keep the node priors the same and perform consensus only on likelihoods [21], [22]. We refer to this approach as Consensus on Likelihoods (CL). The advantage of CL is that it can match the centralized estimator's performance. However, if the network becomes disconnected, priors begin to deviate and become different, and then CL methods fail. For those scenarios, one approach is to perform Iterative Conservative Fusion (ICF) on node posteriors [23], [14], [15]. The work in references [24], [18], [25], [26] also falls into this category. They propose different optimization criteria to perform Conservative Fusion (CF) and/or use different iterative CF schemes for distributed state estimation. ICF methods are inherently sub-optimal as a result of their conservative fusion rule that avoids double counting at the expense of down weighting the uncorrelated information.

Recently, researchers have suggested combining ICF and CL methods to benefit from their complementary features [9], [7], [13]. CL can reach a consensus over uncorrelated new information and ICF can handle the correlated prior information. Such Hybrid methods have been shown to outperform pure ICF's performance and remain robust to link failure [13]. Fig. 1 shows how one can benefit from the Hybrid method for a network with intermittent disconnections. As we discuss in Section VI, use of the Hybrid method can improve the performance of the estimation compared to Iterative Covariance Intersection (ICI) for any possible probability of link failure.

Closest to the work presented here is the research by Battistelli et al. [7], [8], [9], which develops and establishes the stability of their 'Hybrid Consensus on Information and Consensus on Measurements' (HCMCI) method for linear and nonlinear dynamical systems. They assume that the network remains connected for all time. The motivation for their algorithm is that CI, though guaranteeing stability for any number of consensus steps (even a single one), has mean-square estimation error performance adversely affected if terminated before consensus is achieved. The reason is that the fusion rule adopts a conservative point of view, assuming the correlation between the estimates coming from different nodes is completely unknown. On the other hand, Consensus on Measurements (CM) avoids any conservative assumption on the correlation by fusing only the novel information, but it does not guarantee stability unless there are sufficiently many consensus steps. Clearly this can be problematic whenever, for reduced communication cost and improved energy efficiency, only few consensus steps can be performed in each sampling interval.

We believe that the analysis and experiments in this article gives another reason to perform HCMCI. This paper, in re-examining that method (which we, for conciseness, dub the Hybrid method), makes the following contributions:

- **Relaxing connectivity assumptions:** We relax the constant connectivity requirement, something that happens more often than not in practical situations, and show that the Hybrid method remains robust to network failures.
- **Proof of convergence and covariance sandwiching property:** We prove the convergence of the iterative procedure in the Hybrid method and establish the performance bounds for covariance of the local estimators.
- **Extensive and insightful empirical evaluations:** We evaluate the method through extensive experiments showing that in practice the Hybrid method always outperforms, by a large margin, ICI on average.

A preliminary version of this research appeared in [13], but the present paper now includes the fuller theoretical treatment. To this end, the analysis in Section V, including the proof of convergence and complexity analysis, is new. We also introduce more realistic evaluation criteria: we compare the Hybrid method with the condition where the sensors communicate their full history (we term this *Full History Sharing* (FHS), detailed in Section V-C). Moreover, a more realistic simulation that assesses the performance of the method, and compares with FHS, is also presented in Section VI-C.

Motivating Example: Fig. 2 provides an example scenario for the method described in this section. Consider an atmospheric dispersion scenario as an example where there are 6 pollutant sources and 8 sensors distributed in the field, connected to each other through a time varying graph. At first all sensors are connected and all the nodes reach a consensus over the field estimate. Later, for an interval of time, we have two disconnected groups. The sensors in each group continue receiving new information and calculate their local estimates on the basis of their available data. After some time the network regains full connectivity and the agents in each group acquire access to the information accumulated in the other group during the disconnection time. As explained earlier, since the priors of the two groups are distinct, simple averaging is no longer applicable, and using Covariance Intersection results in estimates that are too conservative. The question is how to handle the consensus over estimates when agents are connected, during the disconnection time, and after reconnection.

In Section II, the notation used in this article is explained as well as assumptions and system model. Section III discusses some preliminaries on distributed estimation, paving the way for our problem objective and method. The Hybrid method is presented in Section III along with its theoretical performance analysis. We extensively evaluate the method's performance in Section VI.

II. Modeling

We consider a linear motion and observation model for a system evolving in discrete time:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{w}(k), \tag{1a}$$

$$\mathbf{z}(k) = \boldsymbol{H}(k)\boldsymbol{x}(k) + \boldsymbol{v}(k), \tag{1b}$$

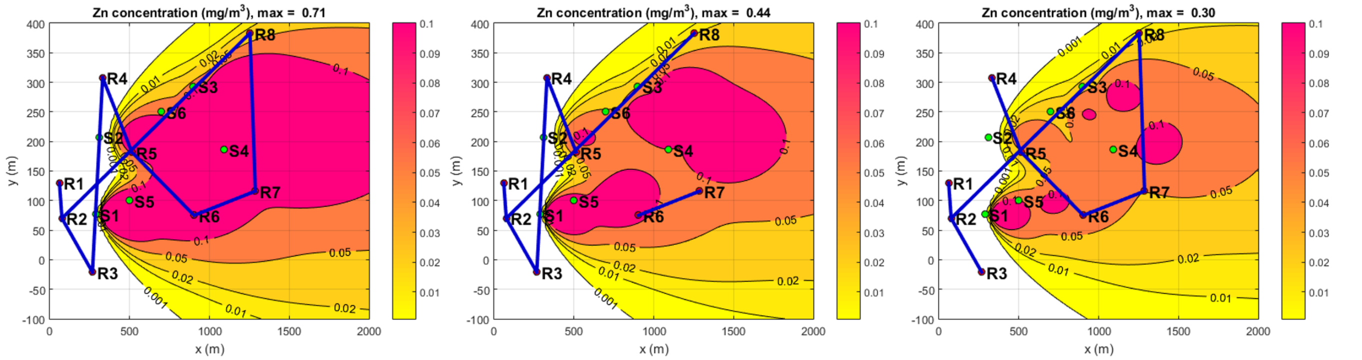


Fig. 2: A motivating example: This is an atmospheric dispersion scenario where 6 pollutant sources and 8 sensors are distributed in the field. They are connected to each other through a time varying graph so that at first all sensors are connected and they remain so for a time interval. Thereafter, due to network interference, we have two disconnected groups. The question is how to handle consensus over estimates after reconnection.

where $\boldsymbol{x}(k) \in \boldsymbol{R}^n$, $\boldsymbol{u}(k) \in \boldsymbol{R}^m$, and $\mathbf{z}(k) \in \boldsymbol{R}^p$ represent state, input, and observation vectors respectively; $\boldsymbol{w}(k) \sim \mathcal{N}(0, \boldsymbol{Q}(k))$ and $\boldsymbol{v}(k) \sim \mathcal{N}(0, \boldsymbol{R}(k))$ represents additive white noise used to model unknown perturbations.

The goal of the general recursive estimation problem is to calculate the posterior probability function $\mathrm{P}(\boldsymbol{x}(k)|\mathbf{z}(k))$ for the system at time k, defined in Eq. 1, given the posterior at step $k-1$, i.e., $\mathrm{P}(\boldsymbol{x}(k-1)|\mathbf{z}(k-1))$. But this paper studies a distributed setting in which the system, in general, does not have access to a monolithic observation vector $\mathbf{z}(k)$. For instance, consider the motivating example depicted in Fig. 2 representing an atmospheric dispersion problem [27]. (As this scenario will also be used for some of our experiments in Section VI, complete details of the model for this problem in the form of Eq. 1 can be found in [28].)

1) Network Topology: Assume that we have N homogeneous agents $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ associated with nodes of a graph. These agents can communicate with each other under a time-varying network topology $G(k) = \langle \mathcal{V}, \mathcal{E}(k)\rangle$ where $\mathcal{E}(k)$ are the set of edges, such that if $(v_i, v_j) \in \mathcal{E}(k)$, it means agents i and j can communicate directly at that time. Neighbors of node v_i are defined as the union of the node v_i and

$$\mathcal{N}_i(k) = \{v_i\} \cup \{v_j \in \mathcal{V} \mid (i,j) \in \mathcal{E}(k)\}. \tag{2}$$

Let $|\mathcal{N}_i(k)|$ denote the cardinality of $\mathcal{N}_i(k)$.

Each agent has a sensory package and a processor on-board. Sensors receive observations in Δt time increments. Every agent's processor and their network connection is fast enough to handle calculations based on message passing every δt units of time. We assume that $\delta t \ll \Delta t$ and that the communication channel is free of delay and error.

2) Observation Model: Each agent's sensor produces noisy observations that are functions of the state of the system. As the decentralized system has N versions of Eq. 1b, the observation model for the i^{th} agent carries an associated subscript:

$$\begin{aligned}\mathbf{z}_i(k) &= \boldsymbol{H}_i(k)\boldsymbol{x}(k) + \boldsymbol{v}_i(k),\\ \boldsymbol{v}_i(k) &\sim \mathcal{N}(0, \boldsymbol{R}_i(k)).\end{aligned}$$

For the atmospheric dispersion problem, observers are receptors which measure the mass of contaminant deposited at their location across time.

3) Observability assumption: We assume that the pair $(\boldsymbol{A}, \boldsymbol{H}_i(k))$ is observable. This means that, under complete network disconnection, individual nodes will produce stable estimates of the system's state.

One expects, and indeed it follows, that the uncertainty in the consensus view of the system's state will decrease as connectivity improves. Next we provide the necessary preliminaries to formalize the notion of this 'consensus view.'

III. Distributed Filtering Preliminaries

Filtering is the process of recursively computing the posterior probability of a random dynamic process $\boldsymbol{x}(k)$ conditioned on a sequence of measurements. The starting point for describing decentralized filtering approaches is the classical centralized case.

A. Centralized Kalman Filter

Under the assumption of Gaussian noise, the Kalman Filter (KF) is the optimal recursive filter for linear state space systems. We use the following notation: $\hat{\boldsymbol{x}} = \boldsymbol{E}(\boldsymbol{x})$ and $\boldsymbol{P}_{\boldsymbol{x}} = \boldsymbol{E}[(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathrm{T}}]$ are the expected value and the covariance of the random variable $\boldsymbol{x}$ respectively. Then, we denote the predicted and estimated mean and covariance at time k by $(\hat{\boldsymbol{x}}^-(k), \boldsymbol{P}_{\boldsymbol{x}}^-(k))$ and $(\hat{\boldsymbol{x}}(k), \boldsymbol{P}_{\boldsymbol{x}}(k))$.

The KF comprises update and prediction steps, both typically using a mean and covariance matrix representation. However, an alternative, the so-called *information form* of the KF, focuses on inverses of the covariances of the Gaussian variables involved. The information form is useful for decentralized filters where it has an intuitive interpretation [1], so we use the equations of this alternative formulation:

$$\boldsymbol{y}(k) = \boldsymbol{P}_{\boldsymbol{x}}^{-1}(k)\boldsymbol{x}(k), \tag{3a}$$
$$\boldsymbol{Y}(k) = \boldsymbol{P}_{\boldsymbol{x}}^{-1}(k), \tag{3b}$$

where $\boldsymbol{y}(k)$ and $\boldsymbol{Y}(k)$ are termed the information vector and information matrix, respectively. The prediction step of the KF can then be written as

$$\boldsymbol{M}(k) = (\boldsymbol{A}^{-1})^T\boldsymbol{Y}(k-1)\boldsymbol{A}^{-1}, \tag{4a}$$
$$\boldsymbol{P}_{\boldsymbol{x}}(k) = \boldsymbol{M}(k) + \boldsymbol{Q}^{-1}(k), \tag{4b}$$
$$\boldsymbol{Y}^-(k) = \boldsymbol{M}(k) - \boldsymbol{M}(k)\boldsymbol{P}_{\boldsymbol{x}}^{-1}(k)\boldsymbol{M}(k), \tag{4c}$$
$$\boldsymbol{y}^-(k) = \boldsymbol{Y}^-(k)\boldsymbol{A}\,\boldsymbol{Y}(k-1)\boldsymbol{y}(k-1). \tag{4d}$$

If this centralized filter were to be implemented in a situation where multiple agents make observations (consider, e.g., the atmospheric dispersion scenario), the agents would transmit their observations to a centralized aggregator. Assuming no network disconnection, the aggregator would perform the steps in Eq. 4. The aggregator's information vector includes a contribution from $\mathbf{z}_j(k)$, the observation of agent j at time k, equal to $\delta \boldsymbol{i}_j(k) = \boldsymbol{H}_j^T(k)\boldsymbol{R}_j^{-1}(k)\mathbf{z}_j(k)$. And the information matrix is updated, reflecting the variance of the agent's observation, with a term $\delta \boldsymbol{I}_j(k) = \boldsymbol{H}_j^T(k)\boldsymbol{R}_j^{-1}(k)\boldsymbol{H}_j(k)$. (In the preceding, the subscript indicates the matrices associated to the agent, and should not be read as selecting columns.) Drawing from all N agents, the aggregate estimate is then:

$$\boldsymbol{y}(k) = \boldsymbol{y}^-(k) + \textstyle\sum_{j=1}^N \delta \boldsymbol{i}_j(k), \tag{5a}$$

$$\boldsymbol{Y}(k) = \boldsymbol{Y}^-(k) + \textstyle\sum_{j=1}^N \delta \boldsymbol{I}_j(k). \tag{5b}$$

This standard formulation is called the Centralized Information Filter (CIF) [29].

However, the assumption of a centralized aggregation process relies on obtaining access to all the information available at each time-step. When each agent can only communicate with its neighbors via transient network links, connectivity may only be sporadic and more sophisticate methods are needed. Next, standard extensions of CIF to decentralized filters that are more suitable for realistic networks are described. These build on the information filter formulation.

B. Decentralized Estimator Designs

1) Consensus-based Estimator: The information filter requires one to have $\delta \boldsymbol{i}(k)$ and $\delta \boldsymbol{I}(k)$. First, we express these entries in terms of averages across agents:

$$\delta \boldsymbol{i}(k) = N \cdot \tfrac{1}{N} \textstyle\sum_{j=1}^N \delta \boldsymbol{i}_j(k), \tag{6a}$$

$$\delta \boldsymbol{I}(k) = N \cdot \tfrac{1}{N} \textstyle\sum_{j=1}^N \delta \boldsymbol{I}_j(k). \tag{6b}$$

Now, were all the agents to obtain $\frac{1}{N}\sum_{j=1}^N \delta \boldsymbol{i}_j(k)$ and $\frac{1}{N}\sum_{j=1}^N \delta \boldsymbol{I}_j(k)$, they could use Eq. 5a–Eq. 5b to calculate an estimate of the system's state. Both expressions represent network-wide averages of quantities that the agents possess locally. Global consensus can be reached over the two factors by performing distributed averaging, so long as all the agents start with the same prior information and N is known. If one could do so, it would yield a state estimate that converges asymptotically to the centralized estimate.

To add some detail: the distributed averaging method of [10] makes minimal assumptions about the network topology and only relies on local information exchange between neighboring nodes of a graph. The method achieves a consensus value that is the average of the initial values of the nodes. It employs an iterative linear consensus filter based on the weights calculated from a Metropolis–Hastings Markov Chain (MHMC). In the equations that follow, we elide the k for the Δt-time-step; the consensus iterations, denoted with an l superscript, operate at the δt timescale. Using a message passing protocol over the communication graph, we can compute $\boldsymbol{x}^{l+1} = \sum_{j=1}^{|\mathcal{N}^l|} \gamma_{ij}^l \boldsymbol{x}_j^l$ to calculate the average of the values on the graph nodes. The weights are computed as follows:

$$\gamma_{ij}^l = \begin{cases} \frac{1}{1+\max\{|\mathcal{N}_i^l|,|\mathcal{N}_j^l|\}} & \text{if } (i,j) \in \mathcal{E}^l, \\ 1 - \sum\limits_{(i,m)\in\mathcal{E}^l} \gamma_{im}^l & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Note that for each node i, the value of γ_{ij}^l depends on the degrees of their neighbors only. Further, an important and established fact (see [10]) is that using MHMC weights for the averaging process will ensure that, after reaching consensus, the estimates will have converged to the centralized estimate. Therefore, given the ideal centralized estimate $(\hat{\boldsymbol{x}}^{\text{CTR}}, \boldsymbol{P}_{\boldsymbol{x}}^{\text{CTR}})$, we have $\hat{\boldsymbol{x}}_i^{\text{MH}} = \hat{\boldsymbol{x}}^{\text{CTR}}$ and $\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{MH}} = \boldsymbol{P}_{\boldsymbol{x}}^{\text{CTR}}$ in the limit. (The superscript was used here, and will be used in the sequel, to differentiate methods used; the mnemonic is: MH denotes Metropolis–Hastings, and CTR for centralized.)

In practice the priors will all differ as a result of network disconnection. In those cases agents have some shared information (from the time they were last connected to each other) but will likely also accumulate new information in periods of disconnection. Their priors will be distinct but correlated after reconnection and, thus, consensus must be handled with care.

2) Covariance Intersection-based Estimator: When the priors differ, distributed averaging alone will not produce consistent estimates. One way of handling such a scenario is using Covariance Intersection (CI). An iterative CI method can be used to reach consensus over the local estimates when the priors differ, either owing to disconnection or termination of the consensus process over-early. In iterative CI [14], the goal is to fuse different estimates of a random variable without having any knowledge about the cross-covariance between such estimates. It solves an optimization problem, updating local estimates iteratively until it reaches consensus. Next, following the discussion in [14]. we describe that optimization problem.

Iterative CI (ICI): Initialization starts with the local estimate for each agent, $[\mathcal{Y}_i^0, y_i^0]$, assigned to be:

$$\mathcal{Y}_i^0 \triangleq \boldsymbol{Y}_i(t_0) + \delta \boldsymbol{I}_i(t_0), \quad y_i^0 \triangleq \boldsymbol{y}_i(t_0) + \delta \boldsymbol{i}_i(t_0).$$

Then, operating at timescale δt, for each iteration, solve for w^* such that

$$\begin{aligned} \omega^* &= \underset{\omega}{\operatorname{argmin}}\ \mathcal{J}\big(\textstyle\sum_{j\in\mathcal{N}_i^l} \omega_j [\mathcal{Y}_j^l]^{-1}\big), \\ s.t. &\quad \textstyle\sum_{j=1}^{|\mathcal{N}_i^l|} \omega_j = 1, \quad \forall j \quad \omega_j \geq 0, \end{aligned} \tag{8}$$

where the optimization objective function, $\mathcal{J}(\cdot)$, is a scalar measure of uncertainty. In general, it is left open as a choice for the system designer (we will consider $\operatorname{trace}(\cdot)$ and $\log\det(\cdot)$, below). Local estimates are then updated for the next iteration via

$$[\mathcal{Y}_i^{l+1}, y_i^{l+1}] = \textstyle\sum_{j\in\mathcal{N}_i^l} \omega_j^* [\mathcal{Y}_j^l, y_j^l]. \tag{9}$$

As discussed in [24], CI and ICI generate estimates that are conservative. Specifically, for the local estimates and the consensus value, this means that $\boldsymbol{E}[\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{ICI}}] = \boldsymbol{E}[\boldsymbol{x} - \hat{\boldsymbol{x}}^{\text{CTR}}] = 0$ and $\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{ICI}} \geq \boldsymbol{P}_{\boldsymbol{x}}^{\text{CTR}}$. Another fact, also shown by [24], is that the ICI method is consistent:

$$\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{ICI}} \geq \boldsymbol{E}[(\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{ICI}})(\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{ICI}})^T]. \tag{10}$$

Consistency implies that the reported covariance matrix, $\boldsymbol{P}^{\text{ICI}}_{\boldsymbol{x}_i}$, is an upper bound of the actual error covariance matrix. The question is: Can we tighten the covariance bound of our estimator without losing consistency? We show that indeed this can be achieved, but care must be taken, lest this seem contradictory. It *is* known that the ICI method is the optimal consistent fusion rule for posteriors when correlation information is unknown. In the information form, one sees that the correlation has inherent structure. Posteriors are mixtures of two parts: priors and new observations, the former contain information shared with other agents, while the latter, importantly, are uncorrelated.

C. Problem Objective

Our goal is to design a recursive decentralized estimator to calculate the local estimate in a manner that is agnostic to the network's topology. (For reasons which become clear shortly, we use 'HYB' to denote the estimator.) We wish to obtain local estimates $\boldsymbol{x}_i^{\text{HYB}}$ and associated covariances $\boldsymbol{P}^{\text{HYB}}_{\boldsymbol{x}_i}$ such that following properties hold:

$$\text{Unbiasedness:} \quad \boldsymbol{E}[\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{ICI}}] = \boldsymbol{E}[\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{HYB}}] = \boldsymbol{E}[\boldsymbol{x} - \hat{\boldsymbol{x}}^{\text{CTR}}] = 0$$
$$\text{Estimate Efficiency:} \quad \mathcal{J}(\boldsymbol{P}^{\text{CTR}}_{\boldsymbol{x}}) \le \mathcal{J}(\boldsymbol{P}^{\text{HYB}}_{\boldsymbol{x}_i}) \le \mathcal{J}(\boldsymbol{P}^{\text{ICI}}_{\boldsymbol{x}_i}) \tag{11}$$

Or, in words, we seek an unbiased estimate whose covariance is an improvement over CI.

IV. Hybrid CI consensus

We outline a Hybrid approach that uses ICI to reach consensus over priors and the MHMC-based consensus filter for distributed averaging of local information updates. The method is summarized in Algorithm 1. We explain the flow of the via a simple scenario with a pair of agents. Generalization to more than two agents is straightforward and follows similar steps.

Suppose two agents observe a dynamic field with state vector $\boldsymbol{x}$; they communicate through a network with time-varying topology. At time t_0, the agents start with priors $[\boldsymbol{y}_1^-(t_0), \boldsymbol{Y}_1^-(t_0)]$ and $[\boldsymbol{y}_2^-(t_0), \boldsymbol{Y}_2^-(t_0)]$ respectively.

By later time t_1, the field has evolved to a new state $\boldsymbol{x}(t_1)$ and agents calculate their own local prediction (line 1 in the algorithm). Then they make observations $z_1(t_1)$ and $z_2(t_1)$, respectively, and compute the local information updates $[\delta\boldsymbol{i}_1(t_1), \delta\boldsymbol{I}_1(t_1)]$ and $[\delta\boldsymbol{i}_2(t_1), \delta\boldsymbol{I}_2(t_1)]$ (lines 2 and 3).

The agents, were they performing ICI, would find a fused estimate

$$\boldsymbol{Y}^{\text{ICI}} = w^{\text{ICI}}(\boldsymbol{Y}_1^- + \delta\boldsymbol{I}_1) + (1 - w^{\text{ICI}})(\boldsymbol{Y}_2^- + \delta\boldsymbol{I}_2), \tag{14}$$

where w^{ICI} is obtained from solving the optimization problem in Eq. 8. In the Hybrid method we do the following:

$$\boldsymbol{Y}^{\text{HYB}} = \underbrace{w^{\text{HYB}}\boldsymbol{Y}_1^- + (1 - w^{\text{HYB}})\boldsymbol{Y}_2^-}_{\substack{\text{ICI to reach} \\ \text{consensus over priors}}} + \underbrace{\delta\boldsymbol{I}_1 + \delta\boldsymbol{I}_2}_{\substack{\text{consensus over the} \\ \text{incremental information}}}. \tag{15}$$

Algorithm 1: Hybrid Method

Input : $[\boldsymbol{y}_j(t_0), \boldsymbol{Y}_j(t_0)]$

1 Use Eq. 4c–Eq. 4d to calculate predicted values $[\boldsymbol{y}_j^-(t_1), \boldsymbol{Y}_j^-(t_1)]$ given $[\boldsymbol{y}_j(t_0), \boldsymbol{Y}_j(t_0)]$

2 Collect local observation $\mathbf{z}_j(t_1)$ and calculate Jacobian and noise covariance $[\boldsymbol{H}_j(t_1), \boldsymbol{R}_j(t_1)]$

3 Calculate the local information update

$$\delta\boldsymbol{i}_j(t_1) = \boldsymbol{H}_j^T(t_1)\boldsymbol{R}_j^{-1}(t_1)\mathbf{z}_j(t_1)$$
$$\delta\boldsymbol{I}_j(t_1) = \boldsymbol{H}_j^T(t_1)\boldsymbol{R}_j^{-1}(t_1)\boldsymbol{H}_j(t_1)$$

4 Initialize consensus variables ($l = 0$)

5 Set

$$[\mathscr{y}_j^0, \mathcal{Y}_j^0] = [\boldsymbol{y}_j^-, \boldsymbol{Y}_j^-](t_1)$$
$$[\overline{\delta\boldsymbol{i}}_j^0, \overline{\delta\boldsymbol{I}}_j^0] = [\delta\boldsymbol{i}_j, \delta\boldsymbol{I}_j](t_1)$$

6 **while** *NOT CONVERGED* **do**

7 BROADCAST$[\mathscr{y}_j^l, \mathcal{Y}_j^l, \overline{\delta\boldsymbol{i}}_j^l, \overline{\delta\boldsymbol{I}}_j^l]$

8 RECEIVE$[\mathscr{y}_k^l, \mathcal{Y}_k^l, \overline{\delta\boldsymbol{i}}_k^l, \overline{\delta\boldsymbol{I}}_k^l] \quad \forall k \in \mathcal{N}_j^l$

9 Collect received data

$$\mathcal{C}_j^l = \{\mathscr{y}^l_{k \in \mathcal{N}_j^l}, \mathcal{Y}^l_{k \in \mathcal{N}_j^l}\} \quad \mathcal{M}_j^l = \{\overline{\delta\boldsymbol{i}}^l_{k \in \mathcal{N}_j^l}, \overline{\delta\boldsymbol{I}}^l_{k \in \mathcal{N}_j^l}\}$$

10 Do one iteration of CI on consensus variables for local prior information $\mathcal{C}_j^l$

$$[\mathscr{y}_j^{l+1}, \mathcal{Y}_j^{l+1}] = \text{CI}(\mathcal{C}_j^l)$$

11 Do one iteration of MHMC on consensus variables for new information $\mathcal{C}_j^l$

$$[\overline{\delta\boldsymbol{i}}_j^{l+1}, \overline{\delta\boldsymbol{I}}_j^{l+1}] = \text{MHMC}(\mathcal{M}_j^l)$$

12 $l \leftarrow l + 1$

13 Calculate the posteriors according to:

$$\boldsymbol{Y}_j(t_1) = \mathcal{Y}_j^l + n_{\text{CG}}\overline{\delta\boldsymbol{I}}_j^l \tag{12}$$
$$\boldsymbol{y}_j(t_1) = \mathscr{y}_j^l + n_{\text{CG}}\overline{\delta\boldsymbol{i}}_j^l \tag{13}$$

return $[\boldsymbol{y}_j(t_1), \boldsymbol{Y}_j(t_1)]$

For an N-agent system with the i^{th} agent having prior Y_i^-, the ICI approach is used to find a consensus over the priors using Eq. 8 recursively. The MHMC approach is used to form the consensus over the new information, i.e., $\sum_{j=1}^{N} \delta\boldsymbol{I}_j$. One cannot do MHMC on Eq. 14 because $\boldsymbol{Y}_1^-$ and $\boldsymbol{Y}_2^-$ differ; note how this contrasts with Eq. 15. Hence, we can use the two pairs $[\mathscr{y}_j^l, \mathcal{Y}_j^l]$ and $[\overline{\delta\boldsymbol{i}}_j^l, \overline{\delta\boldsymbol{I}}_j^l]$ to represent the consensus variables of the i^{th} agent at consensus iteration l for ICI and MHMC processes, respectively.

In line 13 of the algorithm, n_{CG} is the number of agents that form a connected group, which can be determined by assigning unique IDs to the agents and passing them along with the consensus variables. Each agent keeps track of unique IDs it receives and passes them to its neighbors.

V. Analysis

Next, we provide analyses of different aspects of the algorithm.

A. Convergence of the ICI Algorithm

The following demanded that we change notation a little: we denote the consensus iterations via l in parenthesis so as to avoid overloading the superscript. (We continue, as mentioned previously, in dealing with consensus iterations, which happen between two Δt-time-steps.)

Proposition 1. *If the objective $\mathcal{J}(\cdot)$ in Eq. 8 is strictly convex, the ICI process over a connected group in a network is guaranteed to reach a consensus over the priors, i.e., $\exists!\mathcal{Y}_*$, such that $\forall i \lim_{l\to\infty} \mathcal{Y}_i(l) = \mathcal{Y}_*$. The same result holds for the information vector as well.*

Proof. At each iteration l and for each agent j, ICI solves an instance of the optimization problem in Eq. 8. Local variables $\mathcal{Y}_i(l), \forall i \in \{1, \cdots, N\}$ are then updated according to

$$\mathcal{Y}_i^{-1}(l+1) = \sum_{j\in\mathcal{N}_i} \omega_j^* \mathcal{Y}_j^{-1}(l). \tag{16}$$

The definition of the optimization problem in Eq. 8 requires that

$$\mathcal{J}(\mathcal{Y}_i^{-1}(l+1)) \le \mathcal{J}(\mathcal{Y}_j^{-1}(l)), \quad \forall j \in \mathcal{N}_i. \tag{17}$$

Performing ICI is equivalent to a mapping $\mathcal{F}$ that maps the set of local covariance matrices at step l to a new set of covariance matrices at step $l+1$. Defining $\mathcal{I}(l) = [\mathcal{Y}_1^{-1}(l), \cdots, \mathcal{Y}_N^{-1}(l)]$, we can write

$$\mathcal{I}(l+1) = \mathcal{F}\big(\mathcal{I}(l)\big). \tag{18}$$

Next, take the Lyapunov function of the whole network at iteration l to be

$$\mathcal{V}(\mathcal{I}(l)) = \sum_{i=1}^{N} \mathcal{J}(\mathcal{Y}_i^{-1}(l)). \tag{19}$$

If $\mathcal{J}(\cdot)$ is a positive function over the set of Symmetric Positive Definite matrices $\boldsymbol{S}_{++}^N$, then $\forall\, l$, $\mathcal{V}(\mathcal{I}(l)) > 0$. Also, because of Eq. 17, $\mathcal{V}(\mathcal{I}(l+1)) \le \mathcal{V}(\mathcal{I}(l))$. Since $\mathcal{V}$ is monotonically decreasing and bounded below,

$$\lim_{l\to\infty} \mathcal{V}(\mathcal{I}(l)) = \mathcal{V}_*.$$

But convergence of $\mathcal{V}$ does not necessarily mean $\mathcal{I}$ has converged. However, in this case, it turns out to be indeed the case.

Consider the limit set $\Omega = \{\mathcal{I} | \mathcal{V}(\mathcal{I}) = \mathcal{V}_*\}$. If the set Ω consists only of elements $\mathcal{I}$, such that all the the components of any element $\mathcal{I}$ are equal, then, the ICI process becomes stationary, i.e., $\mathcal{F}(\mathcal{I}) = \mathcal{I}$. For any such $\mathcal{I}$ with all components equal, ICI converges to a unique covariance for all nodes. Thus, all we need to show is that Ω cannot have an element such that its components are not all equal.

We proceed by contradiction. Let us assume there is an $\mathcal{I} \in \Omega$ such that the elements of $\mathcal{I}$ are not all equal. Let $\mathcal{I}_m$ denote the m^{th} component of $\mathcal{I}$. Suppose that $\mathcal{I}_j \ne \mathcal{I}_k$, for some k, j. Further, let us assume without loss of generality that: $\mathcal{J}(\mathcal{I}_j) > \mathcal{J}(\mathcal{I}_k)$. Then given any weights ω_j^j, ω_k^j, with $\omega_j^j + \omega_k^j = 1$, we have that $\mathcal{J}(\omega_j^j \mathcal{I}_j + \omega_k^j \mathcal{I}_k) < \omega_j^j \mathcal{J}(\mathcal{I}_j) + \omega_k^j \mathcal{J}(\mathcal{I}_k) < \mathcal{J}(\mathcal{I}_j)$, where the first inequality follows from the strict convexity of $\mathcal{J}(\cdot)$, and the second inequality is due to the convexity of the line segment $[\mathcal{J}(\mathcal{I}_k), \mathcal{J}(\mathcal{I}_j)]$. Using $\mathcal{F}(\mathcal{I})_j$ to denote the j^{th} component of $\mathcal{F}(\mathcal{I})$, from the definition of the optimization inherent in ICI, we see: $\mathcal{J}(\mathcal{F}(\mathcal{I})_j) = \mathcal{J}(\sum_{l\in N_j} \omega_l^{j*} \mathcal{I}_l) \le \mathcal{J}(\omega_j^j \mathcal{I}_j + \omega_k^j \mathcal{I}_k) < \mathcal{J}(\mathcal{I}_j)$, where ω_l^{j*} are the optimal weights resulting from the ICI optimization for node j, and ω_j^j, ω_k^j are the arbitrary weights from before.

Therefore, the Lyapunov function $\mathcal{V}(\mathcal{F}(\mathcal{I})) = \sum_l \mathcal{J}(\mathcal{F}(\mathcal{I})_l) < \sum_l \mathcal{J}(\mathcal{I}_l) = \mathcal{V}(\mathcal{I}) = \mathcal{V}_*$, which contradicts our assumption that $\mathcal{V}_*$ is the lower bound. Thus, any element of Ω has to be such that all its components are equal thus implying that the ICI process converges to a unique covariance at all nodes.

Since the ICI map $\mathcal{F}(\cdot)$ is deterministic, given the initial condition $\mathcal{I}(0)$ is fixed, the set Ω has to be a singleton set, since if there were two elements $\mathcal{I} \ne \mathcal{I}'$ in Ω, it would imply that the ICI map can converge to either of these elements which contradicts the fact that the map is deterministic. □

By establishing strict convexity, the convergence of ICI process is guaranteed by Proposition 1. For instance, $\mathcal{J}(A) \triangleq \log\det(A)$ is strictly convex in its argument [30]. And similarly, $\mathrm{trace}(A)$ is strictly convex.

B. Discussion on Consistency

At the end of Section III-B, we outlined the fact that ICI fusion produces a consistent result (recall discussion of Eq. 10); that fact about the estimate holds regardless of whether fusion is performed on priors or posteriors. Next, we consider the question of consistency for the Hybrid method with an analysis that is generalized to Eq. 12 of Algorithm 1, using the known consistency of ICI.

It suffices to examine the case of two agents, the extension to multiple agents is straightforward.

Suppose the agents' prior estimates are $\boldsymbol{P}_1^-$ and $\boldsymbol{P}_2^-$, respectively. Then, by the definition of the covariance intersection rule,

$$\boldsymbol{P}_{\text{HYB}}^- = \omega \boldsymbol{P}_1^- + (1-\omega)\boldsymbol{P}_2^- > \boldsymbol{P}^-,$$

where ω is the intersection weight and $\boldsymbol{P}^-$ is the true prior covariance. Owing to the ICI consistency:

$$\boldsymbol{Y}_{\text{HYB}} = (\boldsymbol{P}_{\text{HYB}}^-)^{-1} + \delta\boldsymbol{I}_1 + \delta\boldsymbol{I}_2 < (\boldsymbol{P}^-)^{-1} + \delta\boldsymbol{I}_1 + \delta\boldsymbol{I}_2 = \boldsymbol{Y}$$

where $\delta\boldsymbol{I}_j = \boldsymbol{H}_j^T \boldsymbol{R}_j^{-1} \boldsymbol{H}_j$ is the information matrix corresponding to the measurement by the j^{th} agent, $\boldsymbol{Y}$ and $\boldsymbol{Y}_{\text{HYB}}$ represent the true and Hybrid posterior information matrices, respectively. Noting that $\boldsymbol{P} = \boldsymbol{Y}^{-1}, \boldsymbol{P}_{\text{HYB}} = \boldsymbol{Y}_{\text{HYB}}^{-1}$, it follows that $\boldsymbol{P} < \boldsymbol{P}_{\text{HYB}}$. However, $\boldsymbol{P} < \boldsymbol{P}_{\text{HYB}}$ only shows that the filter is conservative. For consistency, we prove in Proposition 2, that the following holds:

$$\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}} \ge \boldsymbol{E}[(\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{HYB}})(\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{HYB}})^T].$$

To understand the consistency of the Hybrid method in practice, including the effects of early termination of consensus, we conduct an evaluation using the Normalized Estimation Error Squared method for realistic scenarios in Section VI-G.

Proposition 2. *The Hybrid filter is consistent i.e.*

$$\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}} \geq \boldsymbol{E}[(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})^T],$$

if the agents in a connected group have reached consensus.

Proof. Let the true covariance of the prior estimate be defined as

$$\bar{\boldsymbol{P}}_{\boldsymbol{x}_i}^{\text{HYB}-} = \boldsymbol{E}[(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}-})(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}-})^T]. \tag{20}$$

We know from Eq. 10 that the ICI process is consistent. So, the convariance (for agent i) $\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-}$, which we obtain from ICI on priors for the Hybrid filter will satisfy: $\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-} \geq \bar{\boldsymbol{P}}_{\boldsymbol{x}_i}^{\text{HYB}-}$.

Using the prior covariance given by ICI, the posterior is computed by the well known Kalman update equations:

$$\hat{\boldsymbol{x}}_i^{\text{HYB}} = \hat{\boldsymbol{x}}_i^{\text{HYB}-} + \boldsymbol{K}(\boldsymbol{z}-\boldsymbol{H}\hat{\boldsymbol{x}}_i^{\text{HYB}-}), \tag{21}$$

$$\hat{\boldsymbol{x}}_i^{\text{HYB}} = \hat{\boldsymbol{x}}_i^{\text{HYB}-} + \boldsymbol{K}\boldsymbol{H}(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}-}) + \boldsymbol{K}\boldsymbol{v},$$

$$\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}} = (\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-}(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})^T + \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T \tag{22}$$

where, $\boldsymbol{K} = \boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-}\boldsymbol{H}^T + \boldsymbol{R})^{-1}$ is the Kalman gain, $\boldsymbol{z}$ is the measurement, and $\boldsymbol{v}$ is the measurement noise. The true covariance of the posterior is given by:

$$\begin{aligned}&\boldsymbol{E}[(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})^T]\\ &\quad= (\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})\boldsymbol{E}[(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}-})(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}-})^T](\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})^T\\ &\qquad+\boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T.\end{aligned}$$

Using Eq. 20 and $\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-} \geq \bar{\boldsymbol{P}}_{\boldsymbol{x}_i}^{\text{HYB}-}$, we get

$$\begin{aligned}\boldsymbol{E}[(\boldsymbol{x}-&\hat{\boldsymbol{x}}_i^{\text{HYB}})(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})^T]\\ &= (\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})\bar{\boldsymbol{P}}_{\boldsymbol{x}_i}^{\text{HYB}-}(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})^T + \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T\\ &\leq (\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}-}(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})^T + \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T\\ &= \boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}}.\end{aligned}$$

Thus,

$$\boldsymbol{E}[(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})(\boldsymbol{x}-\hat{\boldsymbol{x}}_i^{\text{HYB}})^T] \leq \boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}}. \tag{23}$$

□

C. Realistic evaluation criteria

One way to assess performance of a distributed algorithm is to compare its output to that of a centralized estimator with access to all the data. But since, in general, no algorithm subject to network disconnection will fare as well as one not subject to message loss, a better means of comparison ought to be fairer. We consider, instead, the best possible estimator given the network connectivity constraints throughout time. We use the moniker *Full History Sharing* (FHS) for a hypothetical non-recursive method used as a yardstick, which operates as follows: each agent keeps track of its own observations and all the observations ever received (even indirectly) from other agents connected to it. Denote this by $\mathcal{H}_i^t$. If memory and communication constraints are of no concern, at each time-step agents can share their history with each other and update their history according to the shared information. The update rule for $\mathcal{H}_i^t$ is

$$\mathcal{H}_i^t = \bigcup_{\forall j, \mathbf{1}_{i\to j}=1} \mathcal{H}_j^{t-1} \bigcup_{\forall j, \mathbf{1}_{i\to j}=1} \mathbf{z}_j^t \tag{24}$$

where $\mathbf{1}_{i\to j}$ is an indicator function which is 1 when there is a path between node i and j under the current network topology. Obviously $\mathbf{1}_{i\to i} = 1$.

In FHS, at each step, the best possible estimate for each agent is obtained by updating the history and then re-running the filter from scratch. If the network remains connected, the output is equal to the centralized estimator. If the network gets disconnected, FHS gives the best estimate possible.

D. Complexity Analysis

Consider the the problem of distributed estimation of a state vector of dimension n by a system consisting of N agents connected to each other through a network $G = \langle \mathcal{V}, \mathcal{E}(k) \rangle$.

Complexity of the ICI method: The core of CI is a determinant maximization problem and, according to [31], the number of iterations required to solve the optimization is $O(\sqrt{n}f(\epsilon))$ where ϵ is a convergence parameter. For each iteration of the optimization algorithm and for each agent i, cost (considering the objective function $-\log\det(\cdot)$) and gradient calculations are $O(n^3 + |\mathcal{N}_i|n^2)$ and $O(|\mathcal{N}_i|n^2)$ respectively, where $|\mathcal{N}_i|$ is node i's degree. Therefore, the complexity of CI's optimization step is $O(\sqrt{n}(n^3 + |\mathcal{N}_i|n^2))$, where we have suppressed the $f(\epsilon)$ contribution as it is a constant contribution throughout.

Assuming T_{ICI} to be the number of iterations until ICI converges, the computational complexity requirement for each agent can be summarized as

$$O\left(\sqrt{n}(n^3 + |\mathcal{N}_i|n^2)T_{\text{ICI}}\right). \tag{25}$$

ICI relies on passing messages of size $|\mathcal{N}_i|(n^2+n)$ which is independent of the size of the network and only depends on the number of agent i's neighbors.

Complexity of the Hybrid method: For the Hybrid method the cost of doing MHMC consensus should be considered in addition to the ICI steps. Each MHMC consensus iteration updates local covariance in time $O(|\mathcal{N}_i|n^2)$. The convergence times of these algorithms are different in general. Assuming T_{MH} to be the number of iterations until MHMC converges, the computational complexity requirement for each agent can be summarized as

$$O\left(\sqrt{n}(n^3 + |\mathcal{N}_i|n^2)T_{\text{ICI}}\right) + O(|\mathcal{N}_i|n^2 T_{\text{MH}}). \tag{26}$$

The Hybrid method relies on passing messages of size $2|\mathcal{N}_i|(n^2+n)$ for exchanging information with neighbors.

Complexity of the FHS approach: A conservative upper bound for the computational cost is $O(tN^2n^3)$. Even without considering the computational cost of performing the union and the prohibitive memory size and communication requirements for passing messages, the full history estimation cost is larger than the Hybrid method for large t. This makes it a generally impracticable approach, as there is no reason to believe that

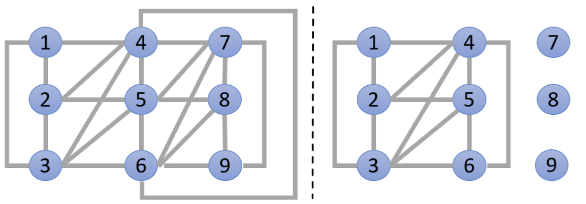


Fig. 3: Topology of the network when all sensors are connected (left) and when sensors 7, 8 and 9 get disconnected from the rest of the group (right).

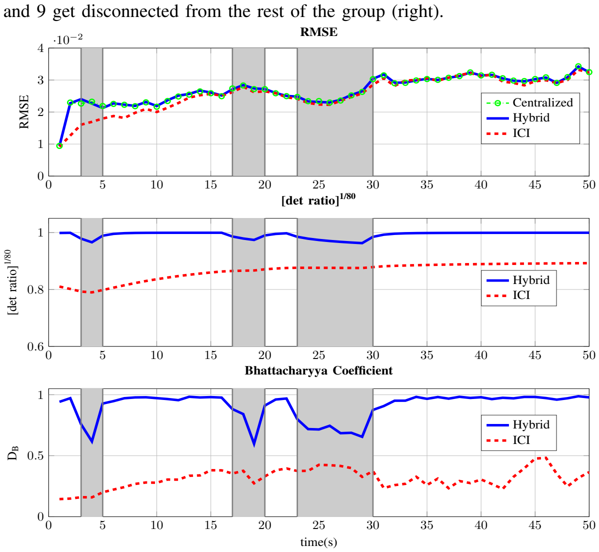


Fig. 4: Comparison of the estimation results using the Centralized Kalman filter, the Iterative Covariance Intersection, and the Hybrid method. The regions shown in grey indicate the time intervals in which sensors 7, 8, 9 are disconnected from the remainder.

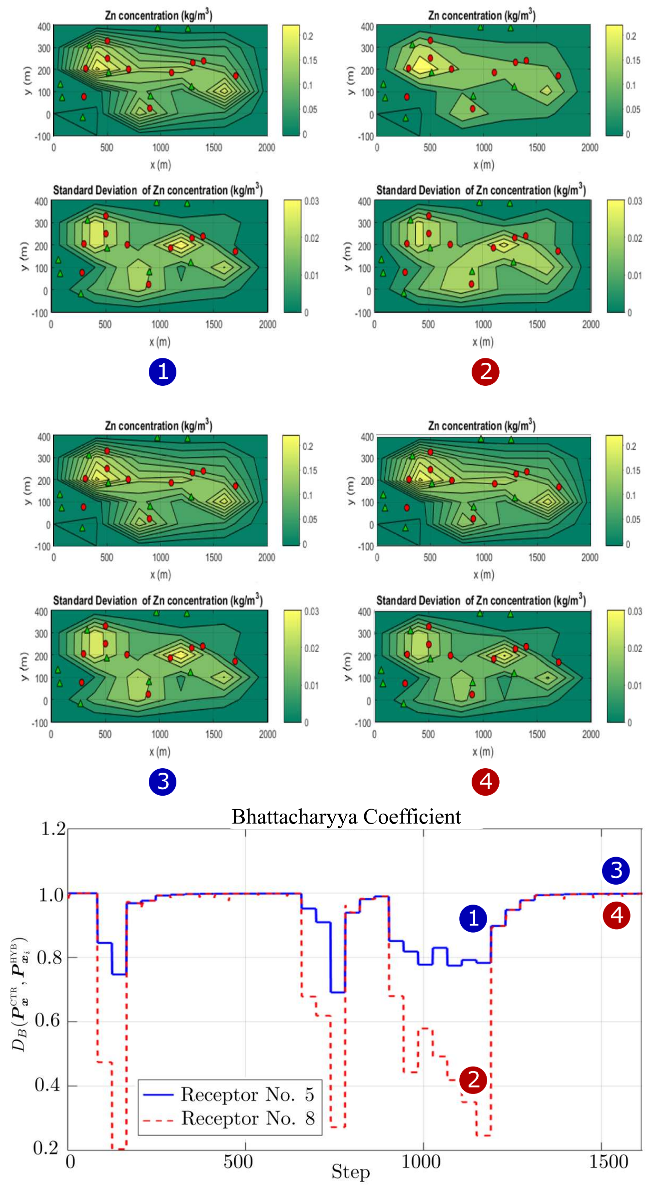


Fig. 5: Estimation performance comparison among sensors 5 and 8. The former is connected to at least 6 nodes at all times, the latter is isolated during network partition. The first row of the images represents the estimate and the second represent the variance. The surface plots give a sense of how the estimates vary: (1) and (2) differ quite distinctly (owing to Receptor No. 8 being disconnected), whereas (3) and (4) are similar (after reconnection). The comparisons are with respect to the Centralized estimator.

tracking would only need to occur for some a priori bounded time.

Memory and message passing requirements to keep the full history also grow linearly in time which finally will make the FHS infeasible for real world applications. We only use the FHS algorithm for comparison purposes as it represents the best achievable performance under the network topology constraints.

VI. Experiments

We performed several experiments on an atmospheric dispersion problem to show the effectiveness of the Hybrid method, evaluate its performance during disconnection and after reconnection and show its scalability, convergence rate and filter consistency. We study a three dimensional problem and, after proper discretizing of its Partial Differential Equation (PDE), we get a system in the form of Eq. 1a.

For our experiments after discretization, the dimension of the state is 80. We assume that there are 10 sources emitting pollutant Zinc (referred to as Zn from now on) into the atmosphere. There are also 9 sensors making noisy measurements of the concentrations of Zn around them. We assume that sensors can communicate with each other through a time varying network which does not remain connected at all times. Sensors receive information only from their immediate neighbors. They all have access to the sources' locations and the source emission is modeled as a white noise process with known covariance. For our experiments on scalability and convergence, we run the filter for 20 time-steps. Since we have multiple sensors, we take the average of the corresponding metric from data given by all the sensors for each time-step and show box plots using samples from different time-steps for a particular case of the respective experiment.

A. *The effect of disconnection on estimation performance*

In this experiment we intend to evaluate the performance of the Hybrid method during the phase where some sensors become disconnected from the rest of the group and get connected again after some interval. The topology of the network takes one of the forms depicted in Fig. 3. The network starts fully connected and starting from time-step 3, sensors

7, 8 and 9 become isolated and remain in this situation for 2 steps, then they are connected back to the rest of the sensors. Similarly, disconnection happens in intervals $[17-20]$ and $[23-30]$.

In order to make a comparison we obtain the estimation result using pure ICI, Hybrid method and also a centralized estimator to see how much of its performance can be recovered. Note that the MHMC consensus cannot be done here due to disconnection. The results are depicted in Fig. 4 where we use three measures to evaluate the estimates.

As it can be seen, the Hybrid method outperforms pure ICI as expected and is able to get performance very close to the centralized estimator results after reconnection. Let $S_B(P_1, P_2)$ be the Bhattacharyya distance [32] between the two distributions P_1 and P_2. We use $D_B(P_1, P_2) = \exp(-S_B(P_1, P_2))$ as a closeness measure between the two distributions. As shown in the figure, the closeness D_B between centralized and distributed estimators drops during the disconnection interval as expected since sensors do not have access to all the information available to the centralized estimator. While the Hybrid method is able to immediately recover after reconnection, pure ICI continues to have lower performance even after reconnection owing to the fact that it calculates conservative upper bounds for the joint covariance matrices.

Fig. 5 takes a closer look at the performance of the Hybrid method and compares the estimation results of sensors 5 and 8 during two different time steps. The horizontal axes represent consensus steps not time. Based on Fig. 3, sensors 5 remains in a group of size 6 during the disconnection periods, whereas sensor 8 is totally isolated at those points in time. The greater difference between centralized and distributed estimates for this sensor can be explained by the fact that it has less information at its disposal. However, after reconnection both sensors are able to converge to the same value, very close to the centralized estimator.

B. Performance analysis and robustness to link failure

In this experiment we evaluate the performance of the Hybrid method in a systematic way to establish its usefulness and robustness to networks with a high probability of link failure. We consider the same system as in the first experiment and simulate it for 50 time steps. At the beginning of each step, a 4 regular graph with 9 nodes is generated and, given a probability of failure for each link, some links in the graph are randomly disconnected. The graph still remains connected some fraction of the times, depending on the regularity degree and probably of link failure. However, with decreasing degree or increasing probability of failure, the network becomes disconnected, more often than not.

In our experiment, for $p \geq 0.2$, consensus methods no longer always succeed since there is almost certainly a case where the network becomes disconnected at some point in time.

We ran the Hybrid method for 50 steps, for each probability of link failure and compared its performance with the ideal centralized result (which is obtained by assuming full connectivity at all times). The performance is evaluated by calculating the average value for Bhattacharyya distance and determinant ratio

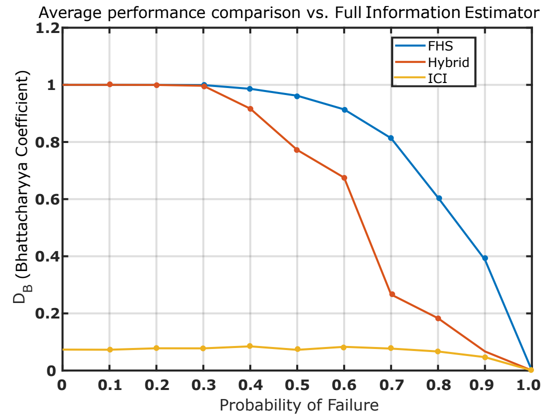


Fig. 6: The estimates of various filters compared against against a centralized estimator that does not suffer from communication failures. The differences are quantified via the Bhattacharyya coefficient.

measure at all steps and for all sensors. Based on Fig. 1, for the case considered in this experiment, the Hybrid distributed estimator performs very similarly to the ideal centralized one for $p \in [0.0, 0.4]$, drastically outperforming pure ICI all the time. This means that in the case considered here, the method can perform almost as well as the ideal estimator for an unreliable network. The Hybrid method recovers the performance of the centralized method when the network is unreliable and outperforms pure ICI substantially (and always does so, as already been established theoretically).

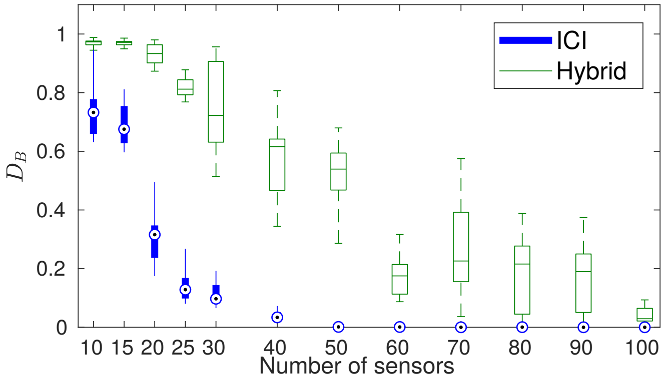
C. Comparison with Full History Sharing method

We performed a comparison with FHS for the atmospheric sampling example. We reduced the dimension of the system from 10^5 to 40 using RPOD (A Randomized Proper Orthogonal Decomposition Technique) [33] and simulated the reduced order system for 80 steps. A comparison of results with that of FHS is shown in Fig. 6. The performance gap between the results of the Hybrid method and FHS is the price of not keeping all the information. The plot shows that, despite the widespread use of ICI in applications, it is inferior to the Hybrid method.

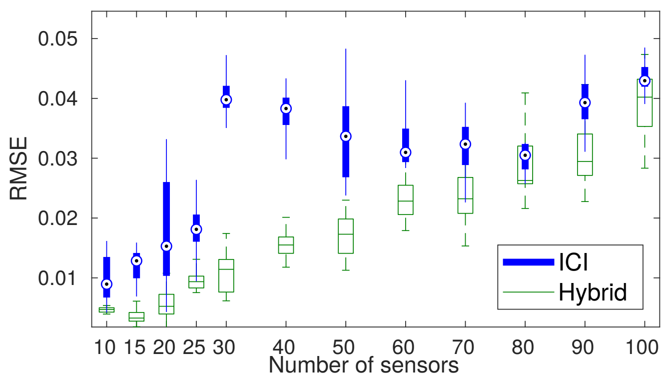
D. Scalability of the Hybrid method

We show that the Hybrid method is scalable by increasing the number of sensors/agents and comparing its performance with ICI . We also show that it is scalable even in the dimension of the state by varying the dimension of the state vector of our system. As shown in Fig. 7, the Hybrid method clearly outperforms ICI not only in estimation performance but also in terms of scalability. Further, the Hybrid estimator is able to match the performance of the FHS estimator for a small network and its performance degrades gracefully as the network size increases.

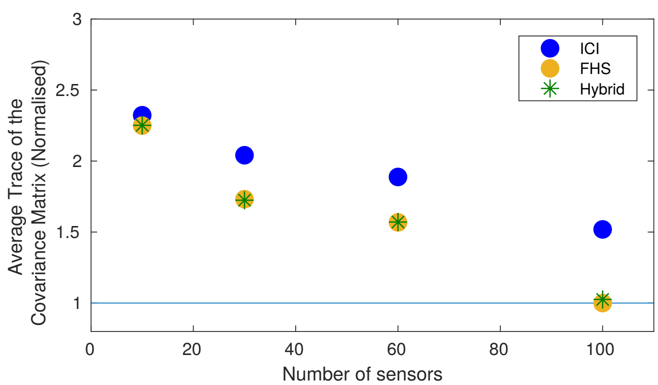
As more sensors are added, more iterations will be needed to reach convergence (this dependency will be considered shortly, when we examine Fig. 8). For both Figs. 7a and 7b, with a fixed number of iterations (60 for all cases), they show the gap between the ideal full history filter and the two algorithms. Because the plots report the *relative* difference from the ideal, it



(a) Performance in terms of Bhattacharyya coefficient (D_B) as network size increases.



(b) Performance in terms of RMSE as network size increases.



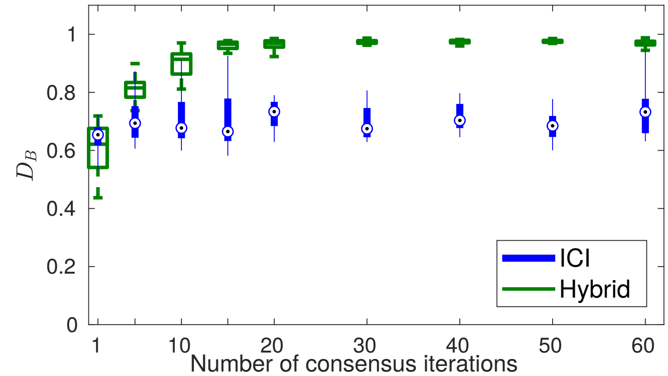
(c) Illustration of how the estimate quality increases with network size though RMSE and D_B metrics show a degradation. This is a trivial result that shows increasing the number of sensors increases the estimate quality.

Fig. 7: Performance comparison of Hybrid and ICI estimators as the number of sensors is varied. D_B of the algorithms is calculated with respect FHS. The data shown are for a state dimension of 80 with a fixed number of 60 consensus iterations and a link failure probability of 0.2.

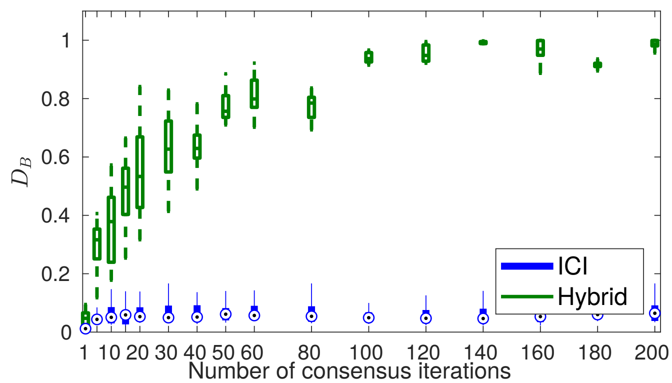
obscures the fact that estimates (for both ICI and Hybrid) with more sensors are better. In other words, though D_B falls moving left to right, the *absolute* estimate quality actually improves. The performance difference between the FHS and the Hybrid estimators with 100 agents is larger than the performance difference between them with 10 agents, and this can be seen in Fig. 7c.

E. Convergence rate of the Hybrid estimator

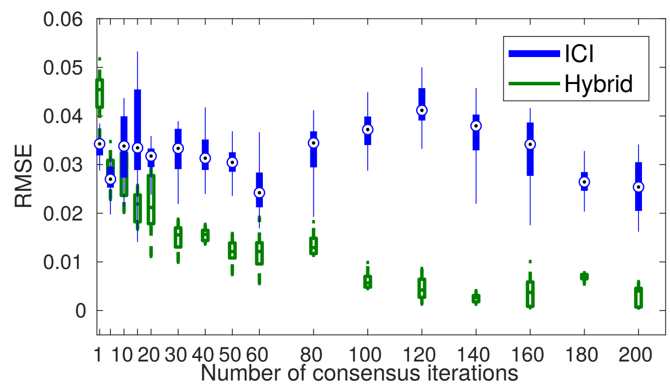
Figs. 8, 9 and 10 show the convergence rate of the Hybrid estimator and how it varies with the network size and connectivity.



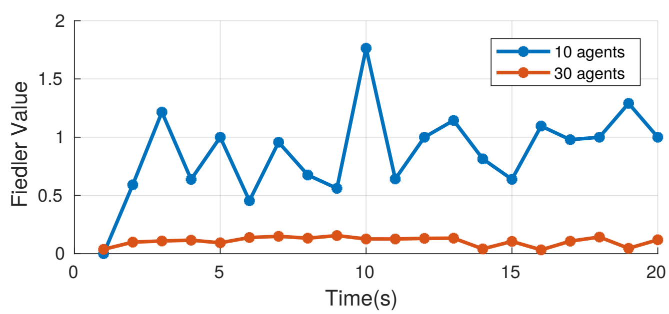
(a) Performance in terms of D_B for 10 agents.



(b) Performance in terms of D_B for 30 agents.



(c) Performance in terms of RMSE for 30 agents.



(d) Fiedler values.

Fig. 8: Convergence rate of Hybrid and ICI estimators. Dimension of the state vector is 80 and the Fiedler values for the experiments are shown in lower subfigure.

As seen, the algorithm converges at an exponential rate. As the network size increases the number of interactions required to converge also increases. The rate of convergence also depends

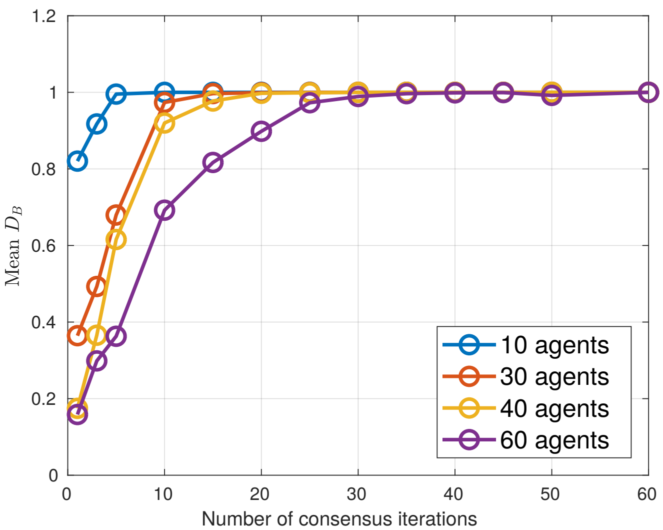
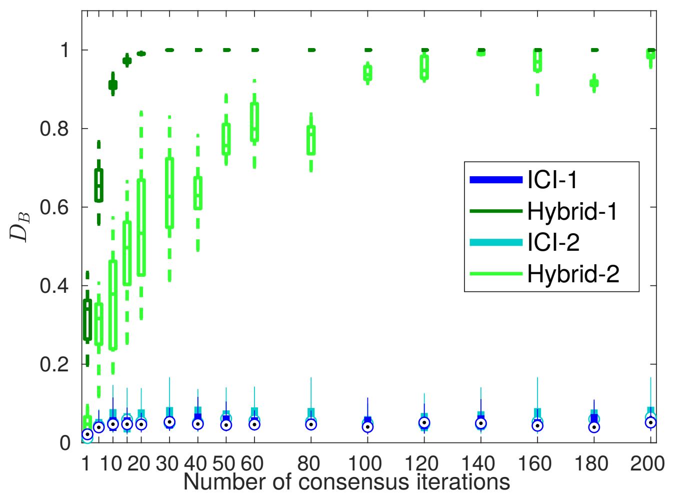
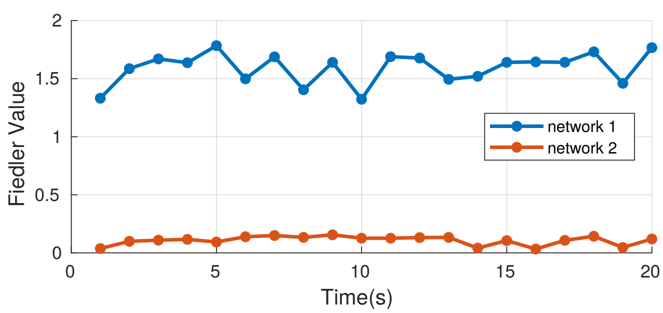


Fig. 9: Convergence of the Hybrid estimator showing exponential behavior.



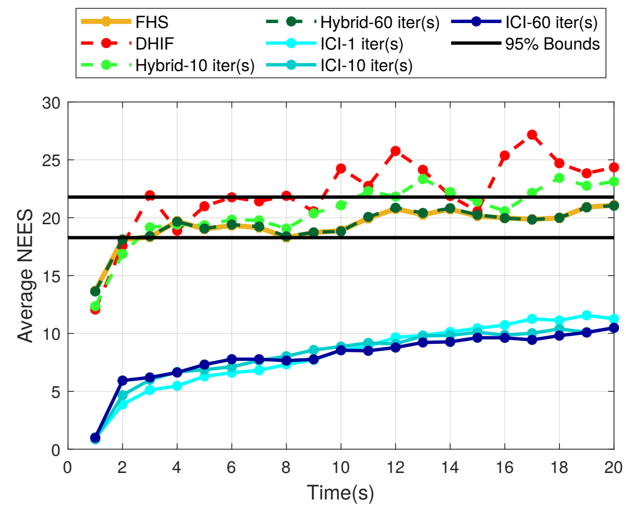
(a) Performance of the algorithms on two networks with different connectivities. ICI-1 and Hybrid-1 denote the performance of the respective algorithms on network 1, and similarly for network 2.



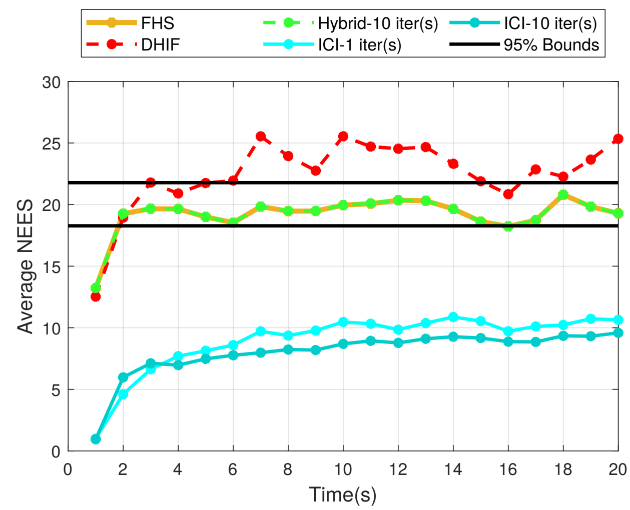
(b) Fiedler values for the networks used in Fig. 10a.

Fig. 10: Comparison of convergence of Hybrid filter in two networks having 30 agents each but with different connectivity. As seen in Fig. 10a, the Hybrid algorithm converges faster on network 1 due to better connectivity.

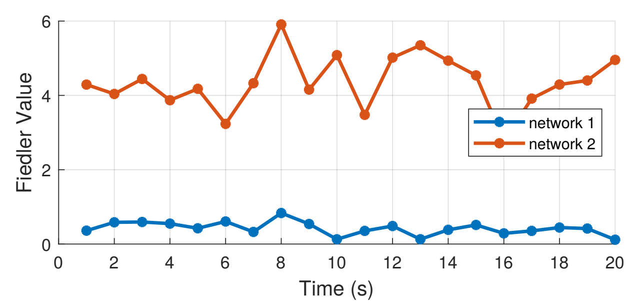
on the connectivity of the network (see Fig. 10) and is the next topic we turn to. Though terminating the algorithm before convergence can sometimes lead to an inconsistent estimate, the iterations can be stopped at any point of time if the application demands it, but by trading away performance.



(a) The NEES test result on network 1.



(b) The NEES test result on network 2.



(c) Fiedler values of the networks used for NEES test.

Fig. 11: The results of NEES tests carried out on a system with 20 states and 15 agents. The number of consensus iterations (iters) done by the filter are also varied to show how it affects the consistency. The Hybrid estimator is also compared with Distributed Hybrid Information Fusion (DHIF) algorithm described in [34]. DHIF is equivalent to the Hybrid algorithm but does consensus only for a single iteration. The data are shown for only one of the agents since the trends are generalisable for the other agents. The corresponding Fiedler values of the networks are also shown.

F. Connectivity of the network

A good indication of network/graph connectivity is the second smallest eigenvalue of the Laplacian matrix of the graph, also called the Fiedler value [35]. The Fiedler value is non-negative with a value strictly greater than zero means that the graph is connected, with higher values indicating better connectivity. The convergence speed of the Hybrid filter is determined by the Fiedler value. The filter converges faster for a network with a high Fiedler value. Fig. 10 shows this effect (and see also, Fig. 11 for further support of this claim). Since our network is prone to changes in its topology at every time-step, the value also changes.

G. Consistency of the Filter

To show the consistency of the filter i.e. $\boldsymbol{P}_{\boldsymbol{x}_i}^{\text{HYB}} \geq \boldsymbol{E}[(\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{HYB}})(\boldsymbol{x} - \hat{\boldsymbol{x}}_i^{\text{HYB}})^T]$, we perform the Normalised Estimation Error Squared (NEES) test [36]. The results of the NEES test carried out for 50 Monte Carlo runs in our atmospheric dispersion problem appears in Fig. 11. As seen, the Hybrid filter is within the 95% bounds of NEES when it has converged or when close to convergence, while ICI is below the bounds meaning its estimate is too conservative. It can also be seen how the Fiedler value affects the convergence rate. (In Fig. 11b the Hybrid filter has converged in 10 consensus iterations while in Fig. 11a it has taken 60 iterations.)

VII. Conclusion

In this paper we studied a distributed estimator for dynamic systems in networks with changing topology and those that do not remain connected all the time. Separating the process of consensus for the correlated and uncorrelated information is one key to achieving better performance when compared to Iterative Covariance Intersection. Evaluating the Hybrid method on an 80 dimensional estimation problem showed substantial performance improvement compared to ICI and also the ability to recover after a disconnection interval occurs. As a summary, these results show that the approach first introduced by Battistelli et al. [9] in order to be a method with attractive stability properties, has a much wider range of applicability than considered heretofore. In fact, the empirical results suggest that time varying networks may come to be seen as its *raison d'être*, rather than its original setting.

Acknowledgement

We thank the editors and reviewers for their valuable comments and suggestions. This work was supported in part by the NSF via awards ECCS-1637889 and IIS-1453652, and in part by NSFC project No. 61801213.

References

[1] H. Durrant-Whyte, M. Stevens, and E. Nettleton, "Data fusion in decentralised sensing networks," in *Proceedings of the 4th International Conference on Information Fusion*. Montréal, Quebec, Canada: IEEE, August 2001, pp. 302–307.

[2] M. E. Campbell and N. R. Ahmed, "Distributed data fusion: Neighbors, rumors, and the art of collective knowledge," *IEEE Control Systems*, vol. 36, no. 4, pp. 83–109, 2016.

[3] N. R. Ahmed, S. J. Julier, J. R. Schoenberg, and M. E. Campbell, "Decentralized bayesian fusion in networks with non-gaussian uncertainties," in *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, F. Hassen and K. Iniewski, Eds. Boca Raton: CRC Press, 2016, ch. 22, p. 383.

[4] H. Zhang, J. Moura, and B. Krogh, "Dynamic field estimation using wireless sensor networks: Tradeoffs between estimation error and communication cost," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2383–2395, 2009.

[5] A. Simonetto, T. Keviczky, and R. Babuška, "Distributed nonlinear estimation for robot localization using weighted consensus," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. Anchorage, AK, USA: IEEE, May 2010, pp. 3026–3031.

[6] W. W. Whitacre and M. E. Campbell, "Cooperative estimation using mobile sensor nodes in the presence of communication loss," *Journal of Aerospace Information Systems*, vol. 10, no. 3, pp. 114–130, 2013.

[7] G. Battistelli, L. Chisci, and C. Fantacci, "Parallel consensus on likelihoods and priors for networked nonlinear filtering," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 787–791, 2014.

[8] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1410–1415, 2015.

[9] G. Battistelli and L. Chisci, "Stability of consensus extended kalman filter for distributed state estimation," *Automatica*, vol. 68, pp. 169–178, 2016.

[10] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*. Boise, ID, USA: IEEE, April 2005, pp. 63–70.

[11] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *IEEE Conference on Decision and Control (CDC)*. Seville, Spain: IEEE, December 2005, pp. 8179–8184.

[12] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, 2013.

[13] A. Tamjidi, S. Chakravorty, and D. Shell, "Unifying consensus and covariance intersection for decentralized state estimation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea: IEEE, October 2016, pp. 125–130.

[14] Y. Wang and X. Li, "Distributed estimation fusion with unavailable cross-correlation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 259–278, 2012.

[15] J. Hu, L. Xie, and C. Zhang, "Diffusion kalman filtering based on covariance intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891–902, 2012.

[16] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proceedings of the 5th International Conference on Information Fusion.*, vol. 2. Annapolis, MD, USA: IEEE, July 2002, pp. 901–904.

[17] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[18] D. Casbeer and R. Beard, "Distributed information filtering using consensus filters," in *American Control Conference (ACC)*. St. Louis, MO, USA: IEEE, June 2009, pp. 1882–1887.

[19] G. G. Rigatos, *Modelling and control for intelligent industrial systems*. Berlin, Germany: Springer, 2011.

[20] L. Mao and D. W. Yang, "Distributed information fusion particle filter," in *In Proceedings of the 6th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1. Hangzhou, China: IEEE, Aug 2014, pp. 194–197.

[21] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4334–4349, 2012.

[22] W. Li and Y. Jia, "Distributed consensus filtering for discrete-time nonlinear systems with non-gaussian noise," *Signal Processing*, vol. 92, no. 10, pp. 2464–2470, 2012.

[23] G. Battistelli and L. Chisci, "Kullback–leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[24] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: covariance intersection revisited," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1879–1882, 2002.

[25] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao, "Sequential covariance intersection fusion kalman filter," *Information Sciences*, vol. 189, pp. 293–309, 2012.

[26] A. Cristofaro, A. Renzaglia, and A. Martinelli, "Distributed information filters for mav cooperative localization," in *Distributed Autonomous Robotic Systems*. Berlin, Germany: Springer, 2013, pp. 133–146.

[27] J. M. Stockie, "The mathematics of atmospheric dispersion modeling," *Siam Review*, vol. 53, no. 2, pp. 349–372, 2011.

[28] A. Tamjidi, "Unifying consensus and covariance intersection for decentralized state estimation," Master's thesis, Texas A&M University, USA, 2017.

[29] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley and Sons, Inc., 2006.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. USA: Cambridge University Press, 2004.

[31] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM journal on matrix analysis and applications*, vol. 19, no. 2, pp. 499–533, 1998.

[32] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: the indian journal of statistics*, pp. 401–406, 1946.

[33] D. Yu and S. Chakravorty, "A randomized proper orthogonal decomposition technique," in *Proceedings of American Control Conference (ACC)*. Chicago, IL, USA: IEEE, July 2015, pp. 1137–1142.

[34] S. Wang and W. Ren, "On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1300–1316, 2018.

[35] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[36] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. USA: John Wiley and Sons, Inc., 2001.

VIII. Appendix

A. Properties of ICI weights

ICI iteration for each agent i starts by making a convex combination of $\mathcal{Y}_i(0) \triangleq \boldsymbol{Y}_i^{\text{ICI}}(0) + \delta\boldsymbol{I}_i(0), \forall i \in \mathcal{N}_i$. In the second iteration,

$$\mathcal{Y}_{j_2}^{\text{ICI}}(1) = \sum_{j_1 \in \mathcal{N}_{j_2}} \omega_{j_1}^{\text{ICI}}(1)\mathcal{Y}_{j_1}^{\text{ICI}}(1) \tag{27}$$

$$= \sum_{j_1 \in \mathcal{N}_{j_2}} \omega_{j_1}^{\text{ICI}}(1)\Big[\sum_{j_0 \in \mathcal{N}_{j_1}} \omega_{j_0}^{\text{ICI}}(0) \Big[Y_{j_0}^{\text{ICI}}(0) + \delta\boldsymbol{I}_{j_0}(0)\Big]\Big] \tag{28}$$

$$= \sum_{j_1 \in \mathcal{N}_{j_2}}\sum_{j_0 \in \mathcal{N}_{j_1}} \omega_{j_1}^{\text{ICI}}(1)\omega_{j_0}^{\text{ICI}}(0)Y_{j_0}^{\text{ICI}}(0) + \sum_{j_1 \in \mathcal{N}_{j_2}}\sum_{j_0 \in \mathcal{N}^{j_1}} \omega_{j_1}^{\text{ICI}}(1)\omega_{j_0}^{\text{ICI}}(0)\delta I_{j_0}(0). \tag{29}$$

One can rewrite the ICI iterations as the multiplication of time varying stochastic matrices by the results from the previous iteration. The multiplication of two stochastic matrices is also a stochastic matrix. Therefore, dropping the ICI superscript for better clarity, the following can be said about $\{m,n\}^{\text{th}}$ element of $\mathcal{Y}_i$'s:

$$\mathcal{Y}^{\{m,n\}}(l+1) = \begin{bmatrix} \mathcal{Y}_1^{\{m,n\}}(l+1) \\ \mathcal{Y}_2^{\{m,n\}}(l+1) \\ \vdots \\ \mathcal{Y}_N^{\{m,n\}}(l+1) \end{bmatrix}$$

$$= \begin{bmatrix} w_{1,1}(l) & \cdots & w_{1,N}(l) \\ w_{2,1}(l) & \cdots & w_{2,N}(l) \\ \vdots & \ddots & \vdots \\ w_{N,1}(l) & \cdots & w_{N,N}(l) \end{bmatrix} \begin{bmatrix} \mathcal{Y}_1^{\{m,n\}}(l) \\ \mathcal{Y}_2^{\{m,n\}}(l) \\ \vdots \\ \mathcal{Y}_N^{\{m,n\}}(l) \end{bmatrix}, \tag{30}$$

in which $w_{i,j}(l) \triangleq 0$ if $\{i,j\} \notin \mathcal{E}$ and for the rest of the elements in i^{th} row where $\{i,j\} \in \mathcal{E}$ at least one of them is non-zero and the non-zero elements always sum to one. In a more concise form,

$$\mathcal{Y}^{\{m,n\}}(l+1) = W^G(l)\mathcal{Y}^{\{m,n\}}(l), \tag{31}$$

in which $W^G(l)$ is the graph topology dependent weight matrix for ICI at iteration l. $W^G(l)$ is a stochastic matrix, hence, the ICI process is equivalent to performing a convex combination of priors and local information matrices.

$$\mathcal{Y}^{\{m,n\}}(\infty) = \lim_{l\to\infty}\prod_{l=1}^{\infty} W^G(l)\mathcal{Y}^{\{m,n\}}(0). \tag{32}$$

We have shown that under ICI, all estimates converge to a unique matrix and given that the ICI is equivalent to a convex combination of initial values over all the nodes, we can concluded that

1) The matrix $W^G(\infty) = \prod_{l=1}^{\infty} W^G(l)$ is a stochastic matrix and has an eigen value of 1
2) The corresponding eigen vector for eigen value 1 is a vector of all ones.
3) The ICI estimate is a convex combination of priors and additional information over all the network nodes, i.e., $\exists w = (w_1, \cdots, \omega_n) \in \boldsymbol{R}^N$, where $\forall i, 0 \le \omega_i \le 1$, $\sum_{i=1}^N \omega_i = 1$ and

$$\mathcal{Y}^{\text{ICI}}(\infty) = \sum_{j=1}^{N}\omega_j\mathcal{Y}_j^{\text{ICI}}(0) + \sum_{j=1}^{N}\omega_j\delta\boldsymbol{I}_j(0). \tag{33}$$

Amirhossein Tamjidi received his MS degree in Control Engineering from K.N. Toosi University of Technology, Tehran, Iran, in 2009 and another M.S. in Aerospace Engineering from Texas A&M university in 2017. His areas of interest are Distributed Estimation, motion planning under uncertainty and SLAM.

Reza Oftadeh is a graduate student at the Department of Computer Science and Engineering, Texas A&M university.

Mohamed Naveed Gul Mohamed is pursuing his Ph.D. in Aerospace Engineering at Texas A&M University, where he is also a member of the Estimation, Decision, and Planning Laboratory. He holds a bachelor's degree in Instrumentation and Control Engineering from NIT Trichy, India.

Dan Yu was born in Harbin, China, in 1988. She received the B.S. degree from Harbin Institute of Technology, Harbin, China, in 2010, and the Ph.D. degree from Texas A&M University, College Station, TX, USA, in 2016.

She is currently an Assistant Professor of College of Astronautics at the Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research has been concerned with distributed systems and robotics.

Suman Chakravorty is an Associate Professor of Aerospace Engineering at Texas A&M University. He holds a Ph.D. in Aerospace Engineering from the University of Michigan, Ann Arbor.

Dylan Shell is an Associate Professor in the Dept. of Computer Science and Engineering at Texas A&M University. He received his B.Sc. degree in computational & applied mathematics and computer science from the University of the Witwatersrand, South Africa, and his M.S. and Ph.D. in Computer Science from the University of Southern California. He took a position as Postdoctoral Research Associate in the USC Interaction lab in 2008. His research aims to synthesize and analyze complex, intelligent behavior in distributed systems that exploit their physical embedding to interact with the physical world.