Efficient Recursive Distributed State Estimation of Hidden Markov Models over Unreliable Networks

Amirhossein Tamjidi[1], Reza Oftadeh[2], Suman Chakravorty[1], and Dylan Shell[2]

***Abstract*—We consider a scenario in which a process of interest, evolving within an environment occupied by several agents, is well-described probablistically via a Markov model. The agents each have local views and observe only some limited partial aspects of the world, but their overall task is to fuse their data to construct an integrated, global portrayal. The problem, however, is that their communications are unreliable: network links may fail, packets can be dropped, and generally the network might be partitioned for protracted periods. The fundamental problem then becomes one of consistency as agents in different parts of the network gain new information from their observations but can only share this with those with whom they are able to communicate. As the communication network changes, different views may be at odds; the challenge is to reconcile these differences. The issue is that correlations must be accounted for, lest some sensor data be double counted, inducing overconfidence or bias.**

As a means to address these problems, a new recursive consensus filter for distributed state estimation on Hidden Markov Models (HMMs) is presented. It is shown to be well-suited to multi-agent settings and associated applications since the algorithm is scalable, robust to network failure, capable of handling non-Gaussian transition and observation models, and is, therefore, quite general. Crucially, no global knowledge of the communication network is ever assumed. We have dubbed the algorithm a Hybrid method because two existing pieces are used in concert: the first, Iterative Conservative Fusion (ICF) is used to reach consensus over potentially correlated priors, while consensus over likelihoods, the second, is handled using weights based on a Metropolis Hastings Markov Chain (MHMC). To attain a detailed understanding of the theoretical upper limit for estimator performance modulo imperfect communication, we introduce an idealized distributed estimator. It is shown that under certain general conditions, the proposed Hybrid method converges exponentially to the ideal distributed estimator, despite the latter being purely conceptual and unrealizable in practice. An extensive evaluation of the Hybrid method, through a series of simulated experiments, shows that its performance surpasses competing algorithms.

I. Introduction

Mobile and robotic-sensor networks have many valuable applications and the problem of estimation within such networks has, consequently, been a topic of extensive study in recent years [1], [2], [3]. In a robotic-sensor network, robots carry sensors that make noisy observations of the state of an system of interest. Many tasks require that the agents construct some overall portrayal of the system's state. This requires that the agents fuse their individual information, ideally in some way that forms a cohesive whole. More precisely, we have to devise methods to estimate the state of the system based on collective information of the agents.

[1]Department of Aerospace Engineering, Texas A&M University
[2]Department of Computer Science & Engineering, Texas A&M University
Both Tamjidi and Oftadeh contributed equally to this paper.
Corresponding author email: ahtamjidi@gmail.com

The estimation process is said to be *centralized* if all the agents send their raw observations to a central node that is responsible for calculating an estimate based on the collective information [4]. But this is not always possible owing to link failures as well as bandwidth and energy constraints [5]; nor is it always desirable because doing so also introduces a single point of failure.

The alternative —termed *distributed state estimation (DSE)*— is to adopt a message passing protocol between agents and strive to achieve the same result as the centralized estimation via a distributed process. In order to be viable, if messages are to contain raw information, the fundamental challenge is to identify and account for mutual information before passing a message from one agent to another. To better understand this notice that a node's knowledge of the process being estimated is based on its own observations and also those of the nodes with whom it has communicated in the past. Because, in real problems, network connectivity may change over time, even two nodes who are exchanging messages with each other for the first time, may both already have incorporated the observations of a common third node, perhaps with whom each communicated individually only in the distant past. Fusion for those two nodes is now a delicate matter as their information is correlated; overconfidence would result if the shared provenance of their estimates is not accounted for properly.

Most DSE research in recent years has focused on approaches that rely on consensus methods. The objective then becomes 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 have fueled interest in the approach. These features are important in the applications envisioned for robots employing such algorithms, such as multi-agent localization [6].

Consistent with this recent line of work, the present paper studies a new algorithm for estimation of Hidden Markov Models over an unreliable network, which we dub the Hybrid method. The algorithm's primary feature is that it achieves attractive performance, in terms of estimate quality, across a wide range of network behavior.

The value and innovation of the Hybrid method is perhaps most easily understood by contrasting it with existing approaches visually. Figure 1 shows different estimation methods and illustrates how the proposed Hybrid method compares. The horizontal axis in the diagram is the probability of link failure: $p = 0$ means that two agents that try to establish a communication link will always succeed and $p = 1$ means they will always fail. The vertical axis represents a performance

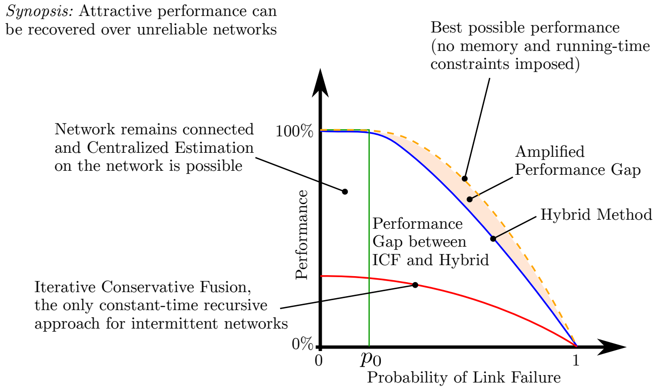


Fig. 1: A graphical synopsis of the paper's contribution. Though a centralized method (in green) yields estimates of highest possible quality, it can only operate when the network remains fully connected for all time. In contrast, Iterative Conservative Fusion (a constant time recursive approach, depicted in red) is practicable with intermittent connectivity, but its conservative nature means that information is diluted, degrading estimation performance. The method studied herein (shown in blue) is a hybrid of both, realizing the "best of both worlds" in that it achieves excellent performance across a wide range of network conditions, including recovering centralized performance in the connected regime.

measure, yet to be defined, that quantifies the similarity of the estimated PMF (Probability Mass Function) to the PMF of an omniscient estimator with access to all observations made by nodes independent from network topology. When the network is connected, this is equivalent to the centralized estimator. However, there is a threshold p_0 (which may in general depend on the network topology) beyond which the centralized estimator cannot operate. Once that threshold has been crossed, the network will likely have more than one connected component and such components will change over time. This is the area where the proposed method has unmistakable superiority over competitors. The threshold p_0 may be small for simple network topologies, resulting in a large area that corresponds to connections of intermittent connectivity.

In the event of intermittent network disconnection, estimator performance inevitably falls below the omniscient estimator. Even if no memory and computational limit is imposed and nodes are allowed to keep the full history of their observations and share such a history with other path connected nodes on the network, there is always an upper bound on the proximity measure. We show that our method's performance approaches the upper bound, resulting in a large performance improvement compared with Iterative Conservative Fusion, an *au courant* method capable of tolerating erratic communications.

A. *The Structure of this Article*

The remainder of the paper is organized as follows. The very next section gives the broad outlines of prior work on distributed state estimation and serves to help familiarize the reader who has come to these problems only lately; it also draws connections, both similarities and points of departure, from the closest existent work. Then, in Section III, the notation used in this paper is explained. That section also identifies the assumptions and describes the system model. Section IV gives some preliminaries on distributed state estimation, paving the way for the presentation of the new Hybrid method. The method itself is presented in Section V and, finally, we evaluate its performance in Section VI.

II. Related Work

A. *Common Modeling Assumptions: A Panoramic View*

Categorizing Distributed State Estimation (DSE) algorithms on the basis of the modeling assumptions they make gives a useful miniature taxonomy of the methods. Any DSE method makes assumptions about one or more of the following aspects: the state (static [7] *vs.* dynamic [6]), the state transition model (linear [8] *vs.* nonlinear [9], [10], [11]), type of noise (Gaussian [7], [8] *vs.* non-Gaussian [12]), topology of the network (constant *vs.* changing [13], [7]), connectivity of the network (persistent [9] *vs.* intermittent [13], [7]), the agents' knowledge of the network topology (global *vs.* local [13], [7], [9]) and, finally, the treatment of mutual information between estimates (exact solution through bookkeeping [1] *vs.* conservative solutions that avoid double counting [14], [15]).

The research on DSE for linear systems with Gaussian noise is extensive (see [8] and [16] for reviews). For nonlinear systems with Gaussian noise, the distributed versions of Extended Kalman Filters (EKF) [17], [9], Extended Information Filters (EIF) [18] and Unscented Kalman Filter (UKF) [19] have been proposed. For nonlinear systems with non-Gaussian noise, different variants of Distributed Particle Filter (DPF) methods have also been studied [20].

B. *Bookkeeping versus Conservative Fusion*

The Channel Filter [1], a classic DSE method, presupposes a directed communication network topology and relies on bookkeeping to make sure no information is double counted during message passing. The Channel Filter can recover the performance of the centralized estimator fully so long as the network is entirely connected and time invariant. A similar bookkeeping-based approach, which relaxes the directed communication graph requirement, was proposed by Bahr, Walter and Leonard [21]. Their method keeps track of the provenance of individual measurements to avoid double counting. The final estimates produced are conservative and consistent approximations of the centralized approach and their method outperforms other conservative fusion DSE approaches that do not perform bookkeeping. However, the basic problem with DSE methods that rely on bookkeeping is their inability to scale. The information being maintained is inherently combinatorial in nature, so they are unsuitable for large-scale networks and their resource requirements (usually for CPU or memory, but possibly communication as well) can be prohibitive even in networks of moderate size.

For dynamic state systems within time-varying networks, the connectivity constraint is a determining factor for choosing the proper DSE method. If the network remains connected, DSE methods can maintain equality of each node's priors and then keep the likelihoods identical by performing consensus on the likelihoods [22], [23]. We refer to this approach as Consensus on Likelihoods (CL). The advantage of CL is that, given sufficient time to reach consensus, it can match the centralized estimator's performance. However, if the network becomes disconnected, or if the consensus steps are limited, the priors start to depart from one another. Once the priors differ across nodes, CL methods will fail.

In the case of disparities between priors owing to network disconnection, the prevailing fix is an approach where the agents perform Iterative Conservative Fusion (ICF) on node posteriors [24], [14], [15]. Such ICF methods have a conservative fusion rule that avoids double counting at the expense of down weighting the uncorrelated information. As a consequence they are inherently sub-optimal. In the case of disparities between priors owing to early termination of a consensus process, [9] and [17] proposed to use a combination of CL and ICF. To justify their method they refer to the complementary features of CL and ICF. They claim that ICF underweights the new information and showed better performance when very few consensus iterations have been executed. On the other hand, CL takes longer to converge but can recover the centralized estimator's performance. Therefore, when the number of consensus iterations are limited, they proposed that the combination would gain the benefits of both.

C. Distributed Calculation of Network Cardinality

For the CL methods to recover the Centralized Estimator's performance fully, they need knowledge of the total number of sensor nodes in the network. This is required because the distributed averaging approach used in CL methods provides the average value of the collective information and, crucially, the sum of the information equals the average value multiplied by the number of nodes.

Fortunately, several distributed methods already exist for computing the cardinality of a network. For example [25] uses the statistical properties of the Bernoulli trials and max consensus to estimate the network cardinality. A review of the prominent methods for distributed node counting can be found in [26] along with a distributed method suitable for dynamic networks. In [27], the authors use randomly generated identifiers (IDs) and propose an algorithm that can estimate the network cardinality with minimal communication cost.

D. Situating this Article's Contribution

Above, the precedence of [9] and [17] in their amalgamation of CL and ICF has been recognized and is openly acknowledged—the present authors were working on the method described herein in order to realize an efficient approach that can cope with conditions of severe network degradation. We encountered their work only after our first publication on the subject [13]. It is worth clarifying the difference between the method in this paper and that of [9] so that, from an application point of view, one should know when to opt for which. At their core both methods follow the same fusion rules for independent and correlated sources of information. This paper describes fusion rules for the discrete case, while [9] details the implementation for continuous linear and non-linear systems with a uni-modal assumption on posterior probability distributions. Save for the typical reasons to prefer a continuous over a discrete representation (i.e., compactness and accuracy), one uses the discrete realization specially when either the (near) Gaussian or uni-modal assumptions no longer hold. Robotics problems violating these assumptions are, of course, known to be abundant.

In essence, our assumptions about the network's behavior differs and the analysis and final results are, consequently, distinct from [9], [17]. (In detail: Proposition 1 gives the quality of network needed to achieve some desired performance—a question that is not meaningful when the network is assumed as in [9] and [17].) Our viewpoint is that network disconnection will inevitably result in unequal priors and using ICF alone will mean that much of the new information, despite being uncorrelated, will be diluted in the consensus process. Handling priors with ICF and new information with CL gives outstanding performance in settings where the communication network is unreliable—so good in fact as to eclipse previously envisioned domains of applicability.

In this paper we analyze the Hybrid method to quantify the relationship between network connectivity and the performance gap with respect to an ideal (yet impractical) distributed estimator. From [9], a complementary aspect is known, namely that the Hybrid method requires few consensus steps to still remain stable under perpetual connectivity. Thus, using our results, the practitioner who has control over the network topology can compare two competing design solutions: balancing the effort to keep the network connected against the loss of some performance with intermittent connectivity. The core proof of this paper, supplementing the discussion in our earlier work [13], helps form a deeper understanding of [9], [17]. One now sees that a hybrid of CL and ICF yields superior performance in networks with either perpetual or intermittent connectivity, for systems where the posteriors are either discrete or continuous and uni-modal, and the motion/observation models are either linear or non-linear. Finally, we point out that the discussion in [9] about the choice of consensus weights when the number of agents in the network is unknown may be applied *mutatis mutandis* to our method as well.

This article is an improved and extended version of the conference paper [28], where the method of [13] was generalized to finite-state systems with non-Gaussian noise. In extending [28] this article has added a mathematical analysis and proof for superiority of the proposed method over ICF. We also show, through systematic examination and extensive simulations, that the performance improvement is significant in situations with any of these traits: large number of agents, significant observation uncertainty, dynamic state systems with several states, and in time-varying networks that face intermittent disconnection. The method handles non-Gaussian noise models, being particularly useful for collaborative tracking and localization. Taken together this supports the claim that the method should be the first choice in many applications.

III. Notation and Model

A. The Network Topology

Assume that we have n homogeneous agents associated with the nodes of a graph. These agents can communicate with one another under a time-varying undirected network topology $G_k = \langle \mathcal{V}, \mathcal{E}_k \rangle$ where $\mathcal{V}$ and $\mathcal{E}_k$ are, respectively, the set of graph nodes and edges at step k. The node corresponding to the i^{th} agent is denoted by v_i. If agents i and j can communicate directly at step k then $(v_i, v_j) \in \mathcal{E}_k$. The set $\overline{\mathcal{N}}_i$ represents the neighbors of node v_i that are connected by an edge to v_i. The set $\mathcal{N}_{i,k} = \overline{\mathcal{N}}_{i,k} \cup \{v_i\}$ will also be used in some of the equations. The set CC_k^i represents the set of agents that are *path-connected* to agent i at step k (the mnemonic being Connected Component).

For a time-varying network, there exist connected component sets (or, more briefly, components) that persist over time. By this we mean that the subset of nodes comprising the component remains constant, though the internal topology of the graph within the component may vary. A component can be uniquely identified by its members, the time of its formation, and its lifetime, i.e., the duration that the set of nodes remains unchanged. Then, at time step k, a network NET_k can be represented by a set of components paired with their formation times:

$$\text{NET}_k = (\text{CC}_k^1, t_{\text{CC}_k^1}), \ldots, (\text{CC}_k^M, t_{\text{CC}_k^M}), \quad M \le |\mathcal{V}|. \tag{1}$$

Agent i is said to be part of a component CC_k^j if $v_i \in \text{CC}_k^j$, where the j is used to denote an entry from the pairs in NET_k, this being an extrinsic view. We will also find it convenient to talk of agent i being part of component CC_k^j, again simply meaning $v_i \in \text{CC}_k^j$. This latter notation refers to the same component (as agent i is only in one component at time k) but it emphasizes a component associated with an individual node, in this particular case, stating that agent i is connected (via some path) to agent j.

The set $T_{cc} = \{t_0, t_1, \ldots\}$ contains the timestamps in which a change occurs in the composition of NET_k. Additionally, let $t_{c,k} = t_{k+1} - t_k$ denote the duration for which all network components comprising NET_k persist. Note that the lifetime of a single component in NET_k can be greater than $t_{c,k}$ if it was formed before t_k or if it continues to exist beyond t_{k+1}.

For an arbitrary set with members $\mathbf{b} = \{b_{i_1}, \cdots, b_{i_s}\}$, the index set $\boldsymbol{I_b} = \{i_1, \cdots, i_s\}$ contains the indices of $\mathbf{b}$'s members (and $s \in \mathbb{N}$). We will use the abbreviated form $\boldsymbol{I}_n = \{1, 2, \cdots, n\}$, and $\boldsymbol{I}_k = \{1, 2, \cdots, k\}$ to index the agents and time steps, respectively.

B. System Model

Consider a finite state HMM specified as follows:

- The HMM has n_s possible states $\mathcal{X} = \{S_1, \cdots, S_{n_s}\}$ and also, there are n_z possible observation symbols $\mathcal{Z} = \{O_1, \cdots, O_{n_z}\}$.
- The random variables $\mathbf{x}_k \in \mathcal{X}$ and $\mathbf{z}_k^i \in \mathcal{Z}$ represent the state at step k and the observation made by agent i at step k, respectively.
- The transition model is an $n_s \times n_s$ matrix written $\mathcal{P}_{k|k-1} \triangleq \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})$. All the agents possess this model.
- Each agent has an observation model, which is an $n_s \times n_z$ matrix written as $\mathrm{p}(\mathbf{z}_k^i|\mathbf{x}_k), i \in \boldsymbol{I}_n$. The observation models of different agents may differ.
- The prior, prediction, and posterior probabilities are $1 \times n_s$ random vectors
$$\pi_{k-1} \triangleq \mathrm{p}\left(\mathbf{x}_{k-1} | \{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_{k-1}}^{i \in \boldsymbol{I}_n}\right),$$
$$\tilde{\pi}_k \triangleq \mathrm{p}\left(\mathbf{x}_k | \{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_{k-1}}^{i \in \boldsymbol{I}_n}, \mathbf{x}_{k-1}\right),$$
$$\pi_k \triangleq \mathrm{p}\left(\mathbf{x}_k | \{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_k}^{i \in \boldsymbol{I}_n}\right),$$
respectively.

The above HMM is a well-defined and useful description for many distributed estimation applications including ones with dynamic state and time-varying observation models. For example, the following transition and observation models can be represented in the above form:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_{k+1}, \mathbf{w}_k) \quad \mathbf{w}_k \sim \mathrm{p}(\mathbf{v}_k), \tag{2}$$
$$\mathbf{z}_{k+1}^i = h^i(\mathbf{x}_{k+1}, \mathbf{v}_k) \quad \mathbf{v}_k \sim \mathrm{p}(\mathbf{v}_k), \tag{3}$$

in which, $\mathbf{w}_k$ and $\mathbf{v}_k$ are random variables representing dynamics and observation noise.

Further, we assume that each agent has a processor and a sensor on-board. Sensors make observations every Δt seconds and the processors and the network can handle calculations based on message passing among agents every δt seconds. We assume that $\delta t \ll \Delta t$. We also assume that the agents exchange their information over a communication channel that is free of both delay and error. Communication links are assumed to be symmetric.

The specification above can be extended to include control inputs but they are omitted as they are not the focus of this paper.

Henceforward $\{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_k}^{i \in \boldsymbol{I}_n}$ is the indexed family of all the observations made by all the agents up to step k. For each agent i, the variable $\mathbf{R}_k^{ij}$ denotes the information that node i receives from node j, its neighbor at time k (i.e., $j \in \overline{\mathcal{N}}_{i,k}$). The set $\mathbf{R}_k^i$ contains all the information that node i has received from its neighbors up to step k and $\mathbf{I}_k^i = \mathbf{R}_k^i \cup \{\mathbf{z}_k^i\}$ represents all the information content that is available to agent i at time k. (In general, in this paper, the information in the variable that bears the superscript i is a version local to the i^{th} agent. Moreover, symbol η with or without any sub/superscript is a normalization constant.)

IV. Distributed State Estimation

In this section we will review some concepts in *Distributed State Estimation* that help us better understand the details of the method developed in the next section. We first define *Recursive State Estimation* in the context of HMMs. Then, we discuss what is meant by *Centralized Estimation* in the context of networked systems, as this notion has been used only informally up till now. We proceed to define a method, within the Consensus on Likelihoods (CL) class, called *Distributed Consensus Based Filtering* that is particular to systems where agents have identical prior information. Given that network disconnection and early stopping of the consensus process yields priors among the agents that are not identical, we review *Conservative Fusion* and its iterative version as a remedy for such cases.

In the context of HMMs, Recursive State Estimation is the process of recursively computing the posterior probability of a random dynamic process $\mathbf{x}_k$ conditioned on a sequence of measurements $\{\mathbf{z}_k^i\}_{k\in \boldsymbol{I}_k}^{i\in \boldsymbol{I}_n}$. Bayesian recursive filtering, in a process with the Markov assumption, has the form

$$\begin{aligned}\mathrm{p}(\mathbf{x}_k|\mathbf{z}_k) &= \frac{1}{\eta}\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{k-1},\mathbf{x}_{k-1}) \qquad (4)\\ &= \frac{1}{\eta}\prod_{i=1}^{n}\mathrm{p}(\mathbf{z}_k^i|\mathbf{x}_k)\int \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})\mathrm{p}(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})\mathrm{d}\mathbf{x_{k-1}},\end{aligned}$$

where $\mathbf{z}_k = \{\mathbf{z}_k^i\}^{i\in \boldsymbol{I}_n}$. Recursive estimation in a sensor network setting for an HMM can be carried out in the following ways:

A. Centralized Estimation

Centralized Estimation (CE) involves a single distinguished node in the network that receives observations $\mathbf{z}_k^{\boldsymbol{I}_n} \triangleq \{\mathbf{z}_k^i\}^{i\in \boldsymbol{I}_n}$ from the rest. The above Bayesian filtering recursion for step k of a finite state HMM consists of first calculating the prediction $\tilde{\pi}_k$ according to

$$\tilde{\pi}_k = \pi_{k-1}\mathcal{P}_{k|k-1}, \qquad (5)$$

then updating via

$$\pi_k = \frac{1}{\eta}\tilde{\pi}_k\mathcal{O}_k, \qquad (6)$$

where $\mathcal{O}_k$ is an $n_s \times n_s$ diagonal matrix of likelihoods, $\mathrm{p}(\mathbf{z}_k^{\boldsymbol{I}_n}|\mathbf{x}_k)$.

Remark 1. Under CE, for a connected component set CC containing n_c nodes, the state Probability Mass Function (PMF) at step k and the initial PMF π_0 are related by

$$^{\text{CE}}\pi_k = \frac{1}{\pi_0 T_{1:k}^{\text{CE}}\mathbf{1}_N}\ \pi_0 T_{1:k}^{\text{CE}}, \qquad (7)$$

where

$$T_{1:k}^{\text{CE}} = \mathcal{P}_{1|0}\mathcal{O}_1\cdots\mathcal{P}_{k|k-1}\mathcal{O}_k. \qquad (8)$$

B. Consensus on Likelihoods

Consensus on Likelihoods (CL) is based on the following insight. In (4), if all agents share the same prior information, they will recover the centralized estimator's performance provided they can reach consensus over the product of measurement probabilities. Distributed averaging methods can be applied here as the nodes need to reach a consensus over the $\log$ of the joint measurement likelihoods (log-likelihood), that is,

$$\tilde{l}_k = \frac{1}{n}\log\prod_{i=1}^{n}\mathcal{O}_k^i = \frac{1}{n}\sum_{i=1}^{n}\log\mathcal{O}_k^i = \frac{1}{n}\sum_{i=1}^{n}\tilde{l}_k^i, \qquad (9)$$

in which $\mathcal{O}_k^i$ is the i^{th} agent's likelihood. Once consensus has been achieved, the updated estimate is

$$\pi_k = \frac{1}{\eta}\overbrace{\underbrace{\pi_{k-1}}_{\text{prior}}\mathcal{P}_{k|k-1}}^{\text{prediction}}\ \underbrace{e^{n\tilde{l}_k}}_{\text{likelihood}}. \qquad (10)$$

Coming to some consensus over likelihoods can be achieved for the discrete state variables using a distributed averaging method based on Metropolis-Hastings Markov Chains (MHMC). To avoid confusion we will use m to indicate consensus iterations throughout this paper. On a communication graph $G\langle\mathcal{V},\mathcal{E}\rangle$ one can use a message passing protocol of the form

$$\begin{aligned}&\psi^i(m+1) = \textstyle\sum_{j=1}^{|\mathcal{N}_i|}\gamma_{ij}(m)\psi^j, \qquad (11)\\ &\text{such that } \sum_j \gamma_{ij}(m) = 1, \forall i \text{ and } \psi^i(0) = \tilde{l}_k^i,\end{aligned}$$

to calculate the average of the values. On the graph nodes in which $d_i(m) = |\mathcal{N}^i|$ is the degree of the node v_i, one sets

$$\gamma_{ij}(m) = \begin{cases}\frac{1}{1+\max\{d_i(m),d_j(m)\}} & \text{if } (i,j)\in\mathcal{E},\\ 1-\sum\limits_{(i,n)\in\mathcal{E}}\gamma_{in} & \text{if } i=j,\\ 0 & \text{otherwise.}\end{cases} \qquad (12)$$

With this message passing protocol

$$\lim_{m\to\infty}\psi^i(m) = \tilde{l}_k.$$

Note that for each node i, the γ_{ij}s only depend on the degrees of its neighboring nodes. As stated earlier, once consensus has been reached over likelihoods, the centralized estimate will be recovered. A prerequisite for this method to work is that the network remains connected, a requirement which is too restrictive for many applications.

Remark 2. For a connected component set CC containing n_c nodes the CL method's likelihood after consensus is equivalent to the likelihood of collective information of the nodes in CC. Introducing the concise notation

$$\mathcal{O}_k^{\{\omega_{1:n_c}\}_k} = \prod_{j=1}^{n_c}[\mathcal{O}_k^j]^{\omega_{j,k}}, \qquad (13)$$

in which $\omega_{j,k}$ is the converged value for the power of $\mathcal{O}_k^j$ in the consensus variable, we see that, if the consensus process converges,

$$\mathcal{O}_k^{\{n_c\omega_{1:n_c}^{\text{CL}}\}_k} = \mathcal{O}_k^{\{n_c\frac{1}{n_c}\}_k} = \mathcal{O}_k. \qquad (14)$$

Even if the topology of the network changes, as long as the nodes that comprise CC remain unchanged, the state PMF at step k and the initial PMF π_0 are related by

$$^{\text{CL}}\pi_k = \frac{1}{\pi_0 T_{1:k}^{\text{CL}}\mathbf{1}_N}\ \pi_0 T_{1:k}^{\text{CL}}, \qquad (15)$$

where

$$T_{1:k}^{\text{CL}} = \mathcal{P}_{1|0}\mathcal{O}_1^{\{n_c\omega_{1:n_c}^{\text{CL}}\}_1}\cdots\mathcal{P}_{k|k-1}\mathcal{O}_k^{\{n_c\omega_{1:n_c}^{\text{CL}}\}_k}. \qquad (16)$$

The expression in (14) guarantees that after convergence, from the same initial condition, the posterior of CL is equal to *CE* and

$$T_{1:k}^{\text{CE}} = T_{1:k}^{\text{CL}} = \mathcal{P}_{1|0}\mathcal{O}_1\cdots\mathcal{P}_{k|k-1}\mathcal{O}_k. \qquad (17)$$

The formal requirement for the above expression to hold is that the consensus process converges with a network dependent rate σ_{CC} and, for δt and Δt defined as before, $\sigma_{\text{CC}}\delta t \ll \Delta t$.

Success of this method is contingent upon the node priors being equal. Next, we discuss the conventional method employed for cases with node priors that may be unequal.

C. Iterative Conservative Filtering

Iterative Conservative Filtering (ICF) is an approach where, instead of putting effort into ascertaining the dependencies between agents' information, a fusion rule is designed to guarantee that no double counting of mutual information can occur. This usually results in the replacement of independent information with some form of approximation that is conservative. Such a treatment dilutes the information available from observations, resulting in performance that is inferior to distributed filters which do not suffer the degradation introduced by this approximation.

Since, in general, *Conservative Approximate Distributed Filtering* relies on fusion rules that combine conservative approximation of local PMFs, we need to clarify what constitutes a conservative approximation for a PMF. Mechanisms of conservative fusion follow conveniently therefrom.

Conservative approximation of a PMF is only possible under certain conditions. Bailey, Julier, and Agamennoni [29] introduced a set of sufficient conditions for a PMF, $\tilde{\mathrm{p}}(\mathbf{x})$, to satisfy in order to be deemed a conservative approximation of a second PMF, $\mathrm{p}(\mathbf{x})$. The conditions are:

(P1) The property of non-decreasing entropy:

$$\mathrm{H}(\mathrm{p}(\mathbf{x})) \leq \mathrm{H}(\tilde{\mathrm{p}}(\mathbf{x}));$$

(P2) The order preservation property:

$$\mathrm{p}(\mathbf{x}_i) \leq \mathrm{p}(\mathbf{x}_j) \text{ iff } \tilde{\mathrm{p}}(\mathbf{x}_i) \leq \tilde{\mathrm{p}}(\mathbf{x}_j),\ \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}.$$

Then Conservative Fusion (CF) of two PMFs can be achieved for two probability distribution functions $\mathrm{p}_a(\mathbf{x}|\mathbf{I}_a)$ and $\mathrm{p}_b(\mathbf{x}|\mathbf{I}_b)$, with the *Geometric Mean Density Rule (GMD)*:

$$\begin{aligned}\mathrm{p}_c(\mathbf{x}) &= \frac{1}{\eta_c}\mathrm{p}_a(\mathbf{x}|\mathbf{I}_a)^{\omega}\mathrm{p}_b(\mathbf{x}|\mathbf{I}_b)^{1-\omega} \\ &= \frac{1}{\eta_c}\mathrm{p}_a(\mathbf{x}|\mathbf{I}_a\setminus\mathbf{I}_b)^{\omega}\mathrm{p}_b(\mathbf{x}|\mathbf{I}_b\setminus\mathbf{I}_b)^{1-\omega}\mathrm{p}_a(\mathbf{x}|\mathbf{I}_a\cap\mathbf{I}_b),\end{aligned} \tag{18}$$

in which, $0 \leq \omega \leq 1$. $\mathbf{I}_b$ and $\mathbf{I}_b$ represent two sources of information. Note that in the above equation the PMFs are raised to the power of ω and multiplied together element-wise. This rule never double counts mutual information, replacing independent components with a conservative approximation. The interesting property of this fusion rule is that it works without the knowledge of the dependence of the two initial PMFs. This rule can also be generalized to more than two PMFs. For example, in the context of this paper, node i calculates a conservative approximation of the centralized estimate and stores it in π^i. The GMD fusion of these estimates, denoted by $\bar{\pi}_k$ is also a conservative approximation of the centralized estimate, π_k.

$$\bar{\pi}_k = \frac{1}{\eta}\prod_{i=1}^{n}\left(\pi_k^i\right)^{\omega_i}, \text{ such that } \quad \textstyle\sum_{i=1}^{n}\omega_i = 1. \tag{19}$$

Note that for the vector π_k^i, the expression $\left(\pi_k^i\right)^{\omega_i}$ implies an element-wise calculation where elements of the vector are raised to the power of ω_i.

Remark 3. Several criteria have been proposed to choose the ω_i. One such criterion is [30]:

$$\bar{\pi} = \arg\min_{\pi}\max_{i}\{\mathcal{D}(\pi\|\pi^i)\}, \tag{20}$$

where the $\mathcal{D}(\pi\|\pi^i)$ is the Kullback-Leibler divergence between π and π^i.

Remark 4. It has been shown in [29] that raising a PMF to some power of $\omega \leq 1$ reduces its entropy. From (19) it can be seen that applying the GMD rule reduces the entropy of the likelihood probabilities that are independent. In general, doing so is undesirable and the likelihood probabilities can be treated separately to avoid this.

Iterative CF (ICF) is achieved as follows. At the first iteration of consensus, $m = 0$, for each agent j, take the current local estimate π_{k-1}^j and calculate the prediction $\tilde{\pi}_k^j$. Initialize the local consensus variable to be

$$\phi^j(0) = \frac{1}{\eta_i}\tilde{\pi}_k^j\mathcal{O}_k^j.$$

Let $\omega = \{\omega_j\}^{j\in\boldsymbol{I}_{\mathcal{N}^i(m)}}$ and find ω^* such that

$$\begin{aligned}&\omega^* = \arg\min_{\omega}\mathcal{J}\Big(\frac{1}{\eta}\prod_{j\in\mathcal{N}^i(m)}\left[\phi^j(m)\right]^{\omega_j}\Big), \\ &\text{and} \sum_{j\in\mathcal{N}^i(m)}\omega_j = 1 \text{ and } \omega_j \geq 0, \quad \forall j,\end{aligned} \tag{21}$$

where η is the normalization constant and $\mathcal{J}(\cdot)$ is an optimization objective function. Specifically it can be entropy $H(\cdot)$ or the criterion in (20). The ϕ^is are then updated locally for the next consensus iteration with

$$\phi^i(m+1) = \frac{1}{\eta^*}\prod\nolimits_{j\in\mathcal{N}^i(m)}\left[\phi^j(m)\right]^{\omega_j^*}. \tag{22}$$

It is straightforward to show that after repeating this process, for all $j \in \mathrm{CC}_k^i$, the local variables $\phi^j(m)$ converge to a unique ϕ^*. Moreover, ϕ^* is a convex combination of the initial consensus variables of all the agents in the set CC_k^i, that is, for all $j \in \boldsymbol{I}_{\mathrm{CC}_k^i}$, where $\boldsymbol{I}_{\mathrm{CC}_k^i}$ is the index set of CC_k^i as defined in notation section,

$$\lim_{m\to\infty}\phi^i(m) = \phi^* = \frac{1}{\eta}\prod\nolimits_{j\in\boldsymbol{I}_{\mathrm{CC}_k^i}}\left[\phi^j(0)\right]^{\omega_j^*} \tag{23}$$

$$= \frac{1}{\eta}\prod\nolimits_{j\in\boldsymbol{I}_{\mathrm{CC}_k^i}}\left[\pi_{k-1}^j\mathcal{P}_{k|k-1}\mathcal{O}_k^j\right]^{\omega_j^*}. \tag{24}$$

To repeat the process iteratively, set $\pi_{k+1}^j = \phi^*, \forall j \in \boldsymbol{I}_{\mathrm{CC}_k^i}$ and repeat the whole process for step $k+1$.

Remark 5. For a connected component CC containing n_c nodes, once the consensus process has converged, we can write the one step estimate update as

$$\begin{aligned}{}^{\mathrm{ICF}}\pi_k &= \tau\left({}^{\mathrm{ICF}}\pi_{k-1}\right) \\ &= \frac{1}{\eta}\prod_{j=1}^{n_c}\left[{}^{\mathrm{ICF}}\pi_{k-1}\mathcal{P}_{k|k-1}\mathcal{O}_k^j\right]^{\omega_{j,k}}.\end{aligned} \tag{25}$$

The expression relating initial PMF and state PMF at step k, unlike previous methods, is a nested expression

$${}^{\mathrm{ICF}}\pi_k = \tau^k(\pi_0) = \tau\big(\tau\big(\cdots\tau(\pi_0)\cdots\big)\big). \tag{26}$$

This shows that under general conditions, even with the same initial PMF, the ICF method will not generate the same estimate over time as CE or CL will. The only exception is the trivial case of a fully disconnected network where, of course, all methods become equivalent.

Remark 6*.* In ICF the nodes' priors are allowed to be different. For the CC described in the previous remark, it only takes one consensus process for all the nodes to have the same prior and to be able to update their state using (25). For a connected set an alternative can be considered: one can use ICF on the priors and, once consensus has been reached, use CL to update the state PMF. This is equivalent to first calculating

$$ {}^{\text{ICF}}\bar{\pi}_0 = \frac{1}{\eta} \prod_{j=1}^{n_c} \left[{}^{\text{ICF}}\pi_0^j \right]^{\omega_{j,0}}, \tag{27} $$

and then using (15) and (16) with $\pi_0 = {}^{\text{ICF}}\bar{\pi}_0$. It is striking that one benefit is that we recover the posterior of CE.

In the previous remark we illustrated how mixing ICF and CL could be beneficial for a connected set of nodes with priors that differ. This is the first indication of the potential for some hybrid between ICF and CL that would be especially useful for networks with intermittent connections, where connected components change over time and it is necessary to handle unequal priors repeatedly. The next section describes such a method. Under the connectivity constraints just mentioned (intermittent communication with connected components that churn) we are able to show that when the lifetime of the connected components in the network is long enough, one can asymptotically recover CE's performance.

V. Hybrid ICF and CL

We propose a hybrid approach that uses ICF to reach consensus over priors and the CL for distributed averaging of local information updates. This is presented in detail as pseudo-code in Algorithm 1 where it is given the designation 'Hybrid method.'

Explanation of the method is aided by having a concrete setting. Imagine a scenario consisting of n agents observing parts of a system at time k and estimating the Markov chain's state $\mathbf{x}_k$ collectively by communicating with one another over a network which has a time-varying topology. Initially the agents start with priors $\{\pi_0^i\}^{i \in \boldsymbol{I}_n}$. At step k the chain transitions to the new state $\mathbf{x}_k$ and the agents calculate their own local prediction $\{\tilde{\pi}_k^i\}^{i \in \boldsymbol{I}_n}$ (line 1 in the algorithm). They then make observations $\{\mathbf{z}_k^i\}^{i \in \boldsymbol{I}_n}$, and compute the local likelihood matrices $\{\mathcal{O}_k^i\}^{i \in \boldsymbol{I}_n}$ (line 1 in the algorithm).

In the rest of the algorithm, the ICF approach is used to reach consensus over the priors using (21) recursively. The CL approach is used to reach consensus over the new information available to agent i from other agents that it is path-connected to, i.e., $\sum_{j \in \boldsymbol{I}_{\text{cc}_k^i}} \tilde{l}_k^i$. In line 12 of the algorithm, $|\text{CC}_k^i|$ is the number of agents that form a connected component with agent i, and can be determined by assigning unique IDs to the agents and passing these IDs along with the consensus variables. Each agent keeps track of the unique IDs it receives, passing them to its neighbors.

Input : π_{k-1}^i

1 Collect local observation $\mathbf{z}_k^i$ and calculate $\mathcal{O}_k^i$ and $\tilde{l}_k^i$
2 Initialize consensus variables:

$$ \phi^i(0) = \pi_k^i, \quad \psi^i(0) = \tilde{l}_k^i $$

3 $m = 0$
4 **while** *not converged* **do**
5   Broadcast$[\psi^i(m), \phi^i(m)]$
6   Receive$[\psi^j(m), \phi^j(m)] \quad \forall j \in \mathcal{N}^i$
7   Collect received data

$$ \mathcal{C}^i(m) = \{\phi^{j \in \mathcal{N}^i}(m)\}, \quad \mathcal{M}^i(m) = \{\psi^{j \in \mathcal{N}^i}(m)\}. $$

8   Do one iteration of ICF on consensus variables for local prior information $\mathcal{C}_m^i$:

$$ \phi^i(m+1) = \text{ICF}\left[\mathcal{C}^i(m)\right]. $$

9   Do one iteration of MHMC on consensus variables for new information:

$$ \psi^i(m+1) = \text{MHMC}\left[\mathcal{M}^i(m)\right]. $$

10     $m \leftarrow m+1$
11 **end**
12 Calculate posteriors according to:

$$ \pi_k^i = \phi^i(m) \mathcal{P}_{k|k-1} e^{|\text{CC}_k^i| \psi^i(m)}. $$

Algorithm 1: The Hybrid method

A. Performance Analysis

To understand the performance of the Hybrid method, we introduce an estimator variant that, though impractical in itself, serves as a useful benchmark for comparison. We use it to conduct an analysis of the comparative performance of the ICF and Hybrid methods.

As was illustrated in Figure 1, beyond a certain point, degradation of the network connectivity causes a catastrophic failure of a centralized estimator. This poses a dilemma if one wishes to analyze the performance of an estimator by comparing its efficiency to an ideal estimator. Comparing against the centralized estimator can hardly be deemed to be meaningful when it must be granted the ability to fuse observations that are inaccessible to a decentralized estimator (e.g., owing to observations being on the opposite side of a network partition). Doing so causes performance measures to be skewed by the unavailability of data rather than the actual estimation process itself.

This motivates consideration of an estimator with performance that is more realistic. As will become apparent shortly, a *Full History Sharing Estimator (FHS)* (see next paragraph) incorporates all the information possible while respecting network topology constraints and, thus, constitutes the proper upper limit for estimator performance.

Full History Sharing Estimator (FHS): Under FHS, at each step k, every agent i has access to the full history of observations of all the agents that it is path connected to at the current step. Then ${}^{\text{FHS}}\pi_k$ is obtained by going back to the initial step, $k=0$, and updating the state PMF sequentially. The update at each step uses all the available observations drawn from the full history. Obviously such calculations quickly become infeasible, but we ignore the computational complexity and only use ${}^{\text{FHS}}\pi_k$ to establish a reference performance. Note that under FHS, even though the whole PMF history is recalculated at each step, the comparison between ${}^{\text{FHS}}\pi_k$ and alternative estimates only involves the PMF at the current point in time.

For our theoretical analysis, we focus on periods of time where the connected components in the network remain unchanged. Note that this assumption allows for change in the network topology so long as it does not result in any change in the connected component sets CC_k. We also make the assumption that consensus processes, of any type, run for enough time to converge for every estimation step.

With these assumptions, the expression relating ${}^{\text{FHS}}\pi_{k+1}$ and the initial PMF, π_0, is

$$ {}^{\text{FHS}}\pi_{k+1} = \frac{1}{\pi_0 T^{\text{FHS}}_{1:k}\mathbf{1}_N}\ \pi_0 T^{\text{FHS}}_{1:k}, \tag{28}$$

where

$$ T^{\text{FHS}}_{1:k} = \mathcal{P}_{1|0}\mathcal{O}_1\cdots\mathcal{P}_{k|k-1}\mathcal{O}_k. \tag{29}$$

Lemma V.1. *Consider the Distributed State Estimation problem of a HMM with a time-varying network topology $G_k = \langle \mathcal{V}, \mathcal{E}_k\rangle$ as described in Section III. At time k_0 let CC^m be the m^{th} connected component containing $|\text{CC}^m| = n_m$ nodes. Further assume that CC^m remains unchanged for the next k steps. Then, during the time $k_0 \le t \le k_0 + k$, and for all the nodes in CC^m, the Hybrid method converges at a geometric rate to the FHS estimator, when the following conditions are satisfied:*

- *The consensus process converges with a network dependent rate σ_{CC^m} . For δt, the consensus update rate, and Δt, the time interval between consecutive observations, we have $\sigma_{\text{CC}^m}\delta t \ll \Delta t$.*
- *The resultant matrix product of the pairs $H(t) \triangleq \mathcal{P}_{t|t-1}\mathcal{O}_t$ is an allowable non-negative matrix, i.e., each row and column of $H(t)$ has at least one positive element.*
- *For a fixed $t_0 \ge k_0$, all the elements of the product chains of both estimators are strictly positive, i.e. $T^{\text{FHS}}_{k_0:t_0} > 0$ and $T^{\text{HYB}}_{k_0:t_0} > 0$.*
- *For a fixed γ, independent of t,*

$$ \frac{\min^+_{i,j} h_{i,j}(t)}{\max_{i,j} h_{i,j}(t)} \ge \gamma > 0, \tag{30}$$

where, $h_{i,j}(t)$ is the (i,j) element of $H(t) \triangleq \mathcal{P}_{t|t-1}\mathcal{O}_t$, and $\min^+$ is the minimum over the positive elements.

Proof. We have already established the main part of the proof by showing that, if the consensus process converges, the inhomogeneous chain of matrix products in (17) and (29) for a connected component are identical. Full history sharing among agents results in a common prior for CC^m as ${}^{\text{FHS}}\pi_{\text{CC}^m,k_0}$. Under the Hybrid method the agents perform conservative fusion of their priors which converges to a unique prior denoted ${}^{\text{HYB}}\pi_{\text{CC}^m,k_0}$. The priors for the two estimators are not the same in general. However, from the moment of connection onwards, as long as CC^m remains unchanged, the inhomogeneous chain of matrix products that results in posterior estimates is equivalent for both methods as shown by (17) and (29), specifically

$$ T^*_{k_0:k} \triangleq T^{\text{FHS}}_{k_0:k} = T^{\text{HYB}}_{k_0:k}. $$

Hence, based on Theorem 3.3 of [31], for which the last three conditions given are required, $T^*_{k_0:k}$ converges to a rank 1 matrix, which consequently renders the initial priors ${}^{\text{FHS}}\pi_{\text{CC}^m,k_0}$, and ${}^{\text{HYB}}\pi_{\text{CC}^m,k_0}$ irrelevant. Therefore the posterior of both estimators converge to the same stationary distribution of $T^*_{k_0:k}$ and, furthermore, they do so at a geometric rate. □

Remark 7. The convergence of $T^*_{k_0:k}$ to a rank 1 matrix is termed weak ergodicity [31], [32]. Moreover, one can use the results of [33] to show that there exists some $\rho_{t_0} < 1$ and $r_{t_0} \le \infty$ so that the decay of the L^1 norm between the posteriors of the two methods is bounded by

$$ \left\| {}^{\text{FHS}}\pi^j_{t_0} T^*_{t_0:t_0+n} - {}^{\text{HYB}}\pi^j_{t_0} T^*_{t_0:t_0+n} \right\|_1 \le r_{t_0}\rho^n_{t_0}. $$

The above expression is the basis for the next lemma. It is also worth pointing out that the geometric nature of this convergence will be clearly visible in the plots showing the method's empirical performance (presented in the following section).

The analysis so far shows that the formation of connected components, and their lifetime, plays an important role in the performance of DSEs. This, in addition to the weak ergodicity property of $T^*_{k_0:k}$, provides practical insight for system designers. One can link the lifetime of a component to L^1 convergence of Hybrid's PMF to FHS's estimates. Also, one might establish some other performance measure for estimate quality and wish to know the requirements on $t_{c,k}$ needed to ensure that the gap between FHS and Hybrid average performance over time is smaller than some desired tolerance. In order to examine these design choices, we need the following definitions.

Let $\mathcal{C}(\cdot)$ be a Lipschitz continuous performance metric that assigns a scalar to a PMF. By definition

$$ \|\mathcal{C}(\pi_1) - \mathcal{C}(\pi_2)\|_1 \le L\|\pi_1 - \pi_2\|_1 \tag{31}$$

where L is the Lipschitz constant.

Lemma V.2. *Let CC^m be a component that was formed at time t_0 and persists for n steps. Let $T^*_{t_0:t_0+n}$ represent the inhomogeneous chain of matrix products that describe the FHS and Hybrid methods for this period. Suppose that FHS and Hybrid priors at time t_0 are ${}^{\text{FHS}}\pi^j_{t_0}$ and ${}^{\text{HYB}}\pi^j_{t_0}$, respectively. For any desired convergence, specified via ϵ_1 such that*

$$ \left\| {}^{\text{FHS}}\pi^j_{t_0} T^*_{t_0:t_0+n} - {}^{\text{HYB}}\pi^j_{t_0} T^*_{t_0:t_0+n} \right\|_1 \le \epsilon_1. \tag{32}$$

CC^m should persist for at least $n = N_{\epsilon_1}$ steps where

$$ N_{\epsilon_1} = (\log_{\rho_{t_0}} \epsilon_1 - \log_{\rho_{t_0}} r_{t_0}) \tag{33}$$

and ρ_{t_0} and r_{t_0} are constants defined in Remark 7.

The lemma is easily proved by taking the logarithm from both sides of inequality (32) and using Remark 7.

Proposition 1. *Consider the behavior of agent j over period of time T, and its connected components for that duration $\mathrm{CC}^j_{t_0}, \mathrm{CC}^j_{t_1}, \ldots \mathrm{CC}^j_{t_m}$. Let the average performance measure of agents j for FHS and Hybrid methods be ${}^{\text{FHS}}J^j$ and ${}^{\text{HYB}}J^j$, respectively. For a given ϵ_1 that satisfies (32) and for a desired $\epsilon_2 > L\epsilon_1$ gap between the average performance of FHS and Hybrid so that*

$$\left\|{}^{\text{FHS}}J^j - {}^{\text{HYB}}J^j\right\|_1 \leq \epsilon_2, \tag{34}$$

the components CC^j_t, $t \in \{t_0, t_1, \ldots, t_m\}$ should persist at least N_{ϵ_2} steps, where

$$N_{\epsilon_2} = \frac{LN_{\epsilon_1}(2-\epsilon_1)}{(\epsilon_2 - L\epsilon_1)} \tag{35}$$

in which N_{ϵ_1} is calculated based on (33) and L is the Lipschitz constant for function $\mathcal{C}(\cdot)$

Proof. By definition, ${}^{\text{FHS}}J^j$ and ${}^{\text{HYB}}J^j$ are

$${}^{\text{FHS}}J^j \triangleq \frac{1}{T}\sum_{t=t_0}^{t_0+T}\mathcal{C}({}^{\text{FHS}}\pi^j_t), \quad {}^{\text{HYB}}J^j \triangleq \frac{1}{T}\sum_{t=t_0}^{t_0+T}\mathcal{C}({}^{\text{HYB}}\pi^j_t). \tag{36}$$

Then we have

$$\left\|{}^{\text{FHS}}J^j - {}^{\text{HYB}}J^j\right\|_1 = \frac{1}{T}\left\|\sum_{t=t_0}^{t_0+T}\mathcal{C}({}^{\text{FHS}}\pi^j_t) - \mathcal{C}({}^{\text{HYB}}\pi^j_t)\right\|_1 \tag{37}$$

$$\leq \frac{1}{T}\sum_{t=t_0}^{t_0+T}\left\|\mathcal{C}({}^{\text{FHS}}\pi^j_t) - \mathcal{C}({}^{\text{HYB}}\pi^j_t)\right\|_1 \tag{38}$$

$$\leq \frac{L}{T}\sum_{t=t_0}^{t_0+T}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1. \tag{39}$$

Now consider that during the time period from t_k to t_{k+1} the j^{th} agent belongs to a connected component whose members are fixed. Then, from Lemma V.1, for some $\rho_{t_k} < 1$ and some $r_{t_k} < \infty$, we have

$$\left\|{}^{\text{FHS}}\pi^j_{t_k}T^*_{t_k:t_k+n} - {}^{\text{HYB}}\pi^j_{t_k}T^*_{t_k:t_k+n}\right\|_1 \leq r_{t_k}\rho^n_{t_k}, \tag{40}$$

for all $n \leq t_{k+1} - t_k$.

For the given $\epsilon_1 > 0$, consider all connected components that agent j belongs to in time interval $[t_0, t_0+T]$ and take $N_{\epsilon_1} = \max_k(\log_{\rho_{t_k}}\epsilon_1 - \log_{\rho_{t_k}} r_{t_k})$. Then, for all $n \geq N_{\epsilon_1}$:

$$\left\|{}^{\text{FHS}}\pi^j_{t_k}T^*_{t_k:t_k+n} - {}^{\text{HYB}}\pi^j_{t_k}T^*_{t_k:t_k+n}\right\|_1 \leq \epsilon_1. \tag{41}$$

Next we denote the duration that the component is connected with $t_{c,k} = t_{k+1} - t_k$ and we introduce a constant that bounds the estimation operation. Let

$$\frac{N_{\epsilon_1}}{t_{c,k}} \leq \delta \implies (t_{k+1} - t_k)\delta \geq N_{\epsilon_1} \tag{42}$$

for all connected periods k and all agents. Then we have

$$\sum_{t=t_0}^{t_0+T}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1 = \sum_k\sum_{t=t_k}^{t_{k+1}}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1. \tag{43}$$

Which can be further expanded into

$$\begin{aligned}\sum_{t=t_k}^{t_{k+1}}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1 &= \sum_{t=t_k}^{t_k+N_{\epsilon_1}}\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\|_1 \\ &\quad + \sum_{t=t_k+N_{\epsilon_1}+1}^{t_{k+1}}\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\|_1 \\ &\leq 2N_{\epsilon_1} + (t_{c,k} - N_{\epsilon_1})\epsilon_1.\end{aligned} \tag{44}$$

The constant appears in the first term of the last expression because the L_1 norm of two probability distributions can never exceed 2.

Then using (42) and (44),

$$\begin{aligned}\frac{1}{T}\sum_{t=t_0}^{t_0+T}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1 &\leq \frac{1}{T}\sum_k(2t_{c,k}\delta + t_{c,k}(1-\delta)\epsilon_1) \\ &= \frac{1}{T}\sum_k t_{c,k}(2\delta + (1-\delta)\epsilon_1) \\ &= (2\delta + (1-\delta)\epsilon_1)\frac{1}{T}\sum_k t_{c,k} \\ &= 2\delta + (1-\delta)\epsilon_1.\end{aligned} \tag{45}$$

Thus,

$$\|{}^{\text{FHS}}J^j - {}^{\text{FHS}}J^j\|_1 \leq L(2\delta + (1-\delta)\epsilon_1). \tag{46}$$

Substituting δ from inequality (42) and using (34) one arrives at N_{ϵ_2} as calculated in (35). □

VI. Experiments

We conduct an analysis of the comparative performance of our method in two ways. First, we examine two case studies (Sections VI-A and VI-B) which, though abstract, are representative of robotic-sensor network applications. Secondly, we carried out experiments where we isolated and controlled various parameters, examining the effect they have on the average performance.

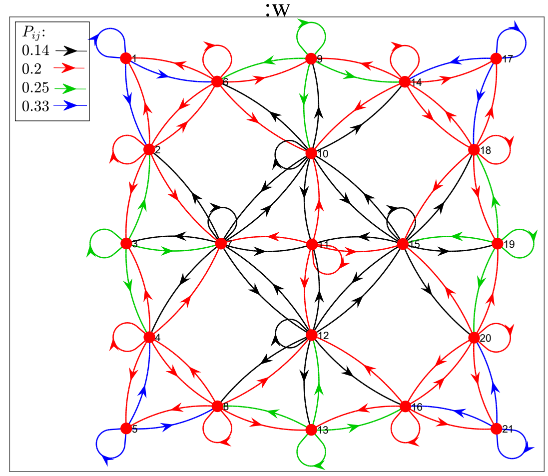


Fig. 2: A schematic of the model used in case study in Section VI-A. The Markov Model has 21 states and is observed by five agents over an unreliable network.

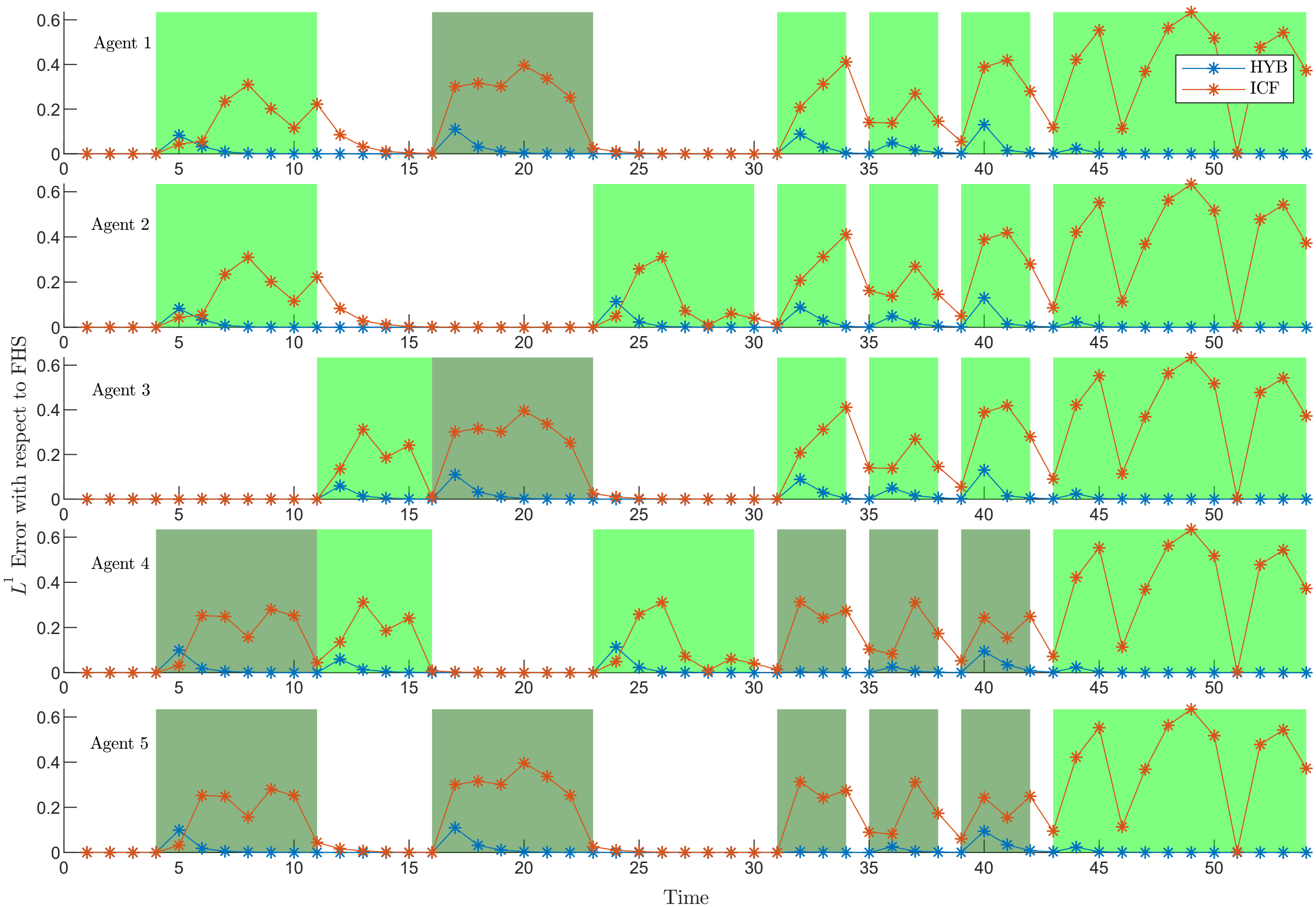


Fig. 3: Performance comparison of the Hybrid method (designated HYB) and ICF on the Markov Model in Figure 2. The green shaded areas mark the lifetime of components and agents with the same shade belong to the same component. Non-shaded areas signify time intervals where the agents are completely isolated from the rest of the network.

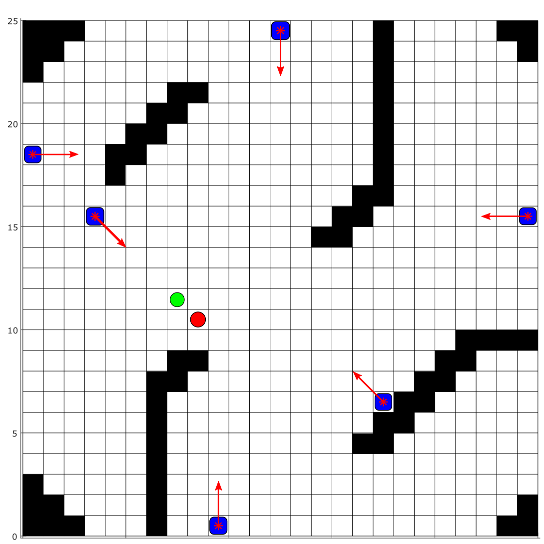


Fig. 4: The grid based map of the environment for the tracking example VI-B. The dark cells are obstacles; blue circles are trackers and the red circle is the ground truth location of the maneuvering target; the green circle depicts an observation made by an agent.

A. Convergence properties in a generic Markov Chain

In the first case study we consider a system consisting of five agents connected to each other through a time-varying network. Agents make observations of the state of a HMM with 21 states. The transition model of the HMM is shown in Figure 2, with transition probabilities represented by color coded arrows. The plots in Figure 3 show the performance of the Hybrid and ICF methods compared to FHS for 55 steps, as connected components form and change. The horizontal axis shows the progression of time; the vertical axis is the difference between estimated PMF and FHS (measured with L^1 norm). The convergence behavior discussed in Remark 7 is directly visible in the Hybrid method and, for this system of moderate size, in most cases convergence takes three steps after the formation of a component. Note also how components with more agents experience faster convergence. One particularly salient instance in Figure 3 is the rapid convergence after step forty-four where the network becomes fully connected.

In contrast, ICF's performance is erratic during connected times possessing exponential convergence only for agents that are disconnected from the rest of the network. This phenomenon can be explained using (26), where we established that, under

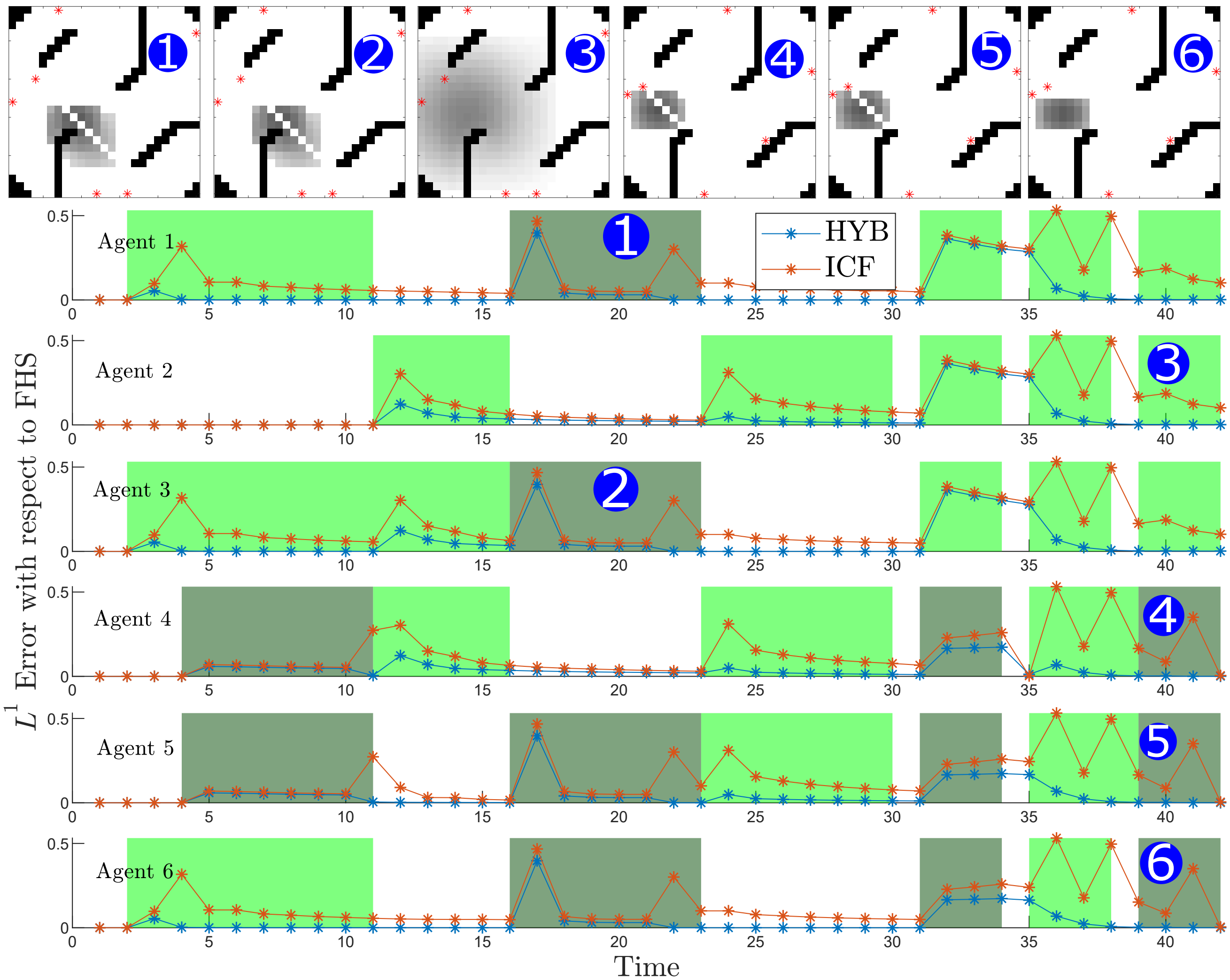


Fig. 5: Estimation performance for the tracking case study (see Figure 4 for the environment and details of the scenario). For each agent its distribution at a single step is shown at the top and is indexed accordingly at its representative time. Non-shaded areas signify time intervals where agents are completely isolated from the rest of the network.

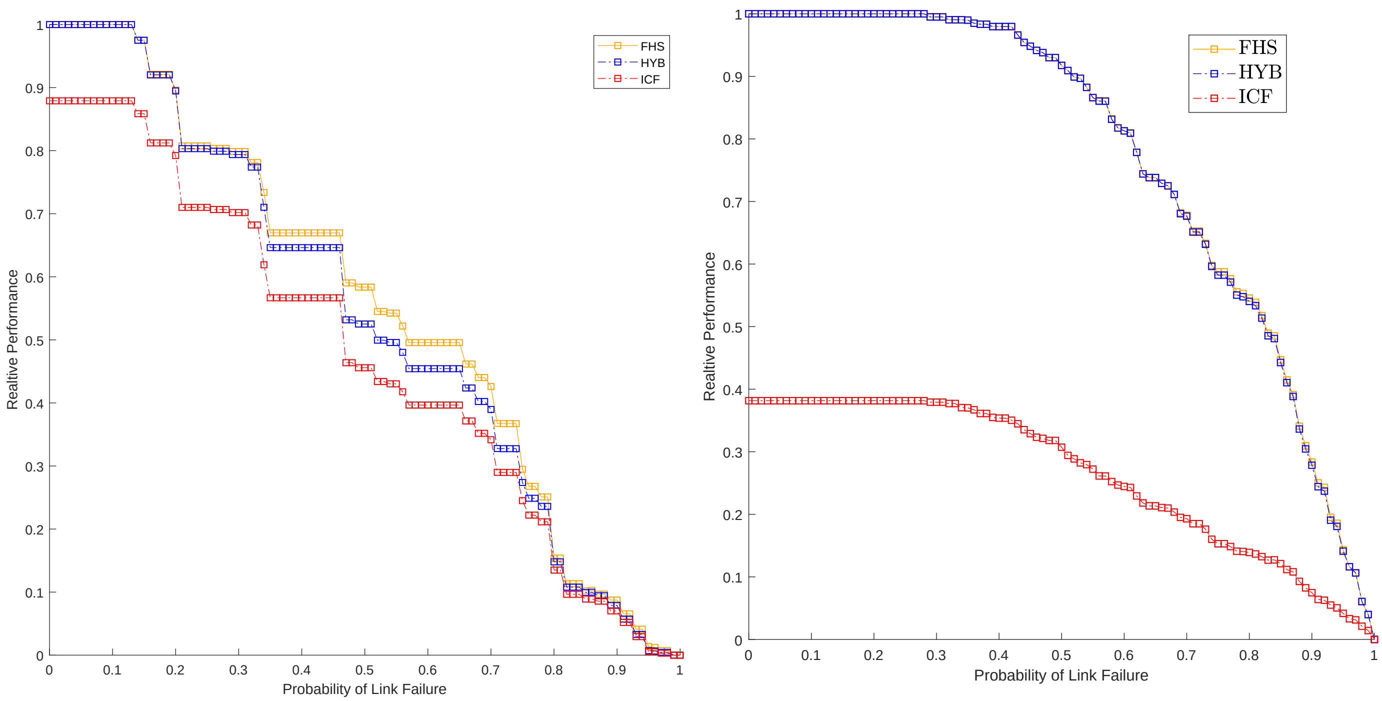
general conditions, for connected components, $T^{\text{FHS}}_{t_k:t_k+n} \neq T^{\text{ICF}}_{t_k:t_k+n}$ except for the trivial case of a component with single agent. The exponential convergence of ICF for agent 2 in $t \in [11, 24]$ is one such case. The convergence in this period is due to the forgetting factor of the HMM.

B. A Tracking Example

Our second case study is concerned with a decentralized target pose estimation problem on a grid using multiple observers connected through a changing network topology. Figure 4 depicts the 2D grid in which a target performs a random walk while six observers are trying to estimate its position. Each white cell is modeled as a single state of our HMM representing the position of the target on the grid. The observers' motion is a deterministic back-and-forth patrolling route; four of them are rooks moving along the borders and the other two are bishops moving diagonally on the grid. In order to detect the target, each observer emits a straight beam perpendicular to its direction of motion as shown in the figure. The beam either hits the target or an obstacle. In the first case, the observer senses the position of the target based on a discrete one dimensional Gaussian distribution over the states that the beam has traversed; otherwise, under the assumption of no false positives, the observer produces a "no target" symbol. (The model has an additional state, which is incorporated into the observation model by setting zero probabilities in the likelihood matrix for those states that beam has traveled through to hit a wall.)

For every Markov transition, each observer carries out its decentralized estimation step for the position of the target, which is shared with other connected observers through the communication network. The network topology varies randomly resulting in the formation of different connected components. However, we assume all communications occur at a higher rate than Markov transition steps, allowing the connected nodes to reach consensus over the shared information.

We evaluate the performance of the Hybrid method during the phase where the agents become disconnected from each other, and are then reconnected after some interval. Similar to the previous case, for purposes of comparison, each agent



(a) Average performance for chessboard tracking example

(b) Average performance for system with three states and twenty observers

Fig. 6: Performance comparison between the Hybrid method and ICF. The horizontal axis is the probability of link failure, moving from left to right shows the network changing from ideal, through fragmentation, to complete failure. The vertical axis is a metric of estimator performance computed as follows: at each time, for every agent, the total variation distance of the estimator's PMF and the output from a hypothetical, omniscient centralized estimator (as if it were operating on a perfect network) is computed. The mean of this is taken over agents and over all times, and then normalized between 0 and 1, where 1 coincides with the fully connected network and 0 the fully disconnected one.

performs three estimation processes. In one instance it uses our Hybrid method to fuse its prior along with the received priors. In the second instance it uses the ICF method to fuse its posterior along with the received posteriors, and the third instance is the FHS method, to give a baseline for comparison. Again, L^1 norm difference is used to make the comparison.

Figure 5 compares the performance of the Hybrid and ICF methods, showing that the proposed method outperforms ICF and is able to recover performance very close to FHS solution after reconnection. Using the same visual presentation as before, the shaded areas mark the lifetime of components and agents with the same shade color belong to the same component. Based on the L^1 distance, both decentralized estimates converge to FHS during the interval of network partition. This is expected, since observers do not have access to each others information and hence, due to the forgetting property of the system, all three estimators become indistinguishable—each separate agent independently performing its own Bayesian update. However, while the Hybrid method is able to start to recover immediately after reconnection, ICF continues with degraded performance even after reconnection. This latter fact is because it ignores the correlations.

C. Focused Performance Evaluation

Next we study the robustness of our method more systematically with respect to network failure. This permits some reflection on the factors that affect the gap between the average performance of our Hybrid method and FHS. The experiments reported in this subsection were performed as follows.

We take the HMM and construct a path connected communication network that is a ring lattice with degree four. This is the base network topology. We then assign a probability of link failure p to all the links in the communication graph and run FHS, HYB, and ICF methods for 50 steps. At each step we randomly disconnect links in the base graph, with probability p and perform the consensus processes on the resulting graph. For π_k^j, the local estimate of agent j at time k, and π_k^*, the estimate from the omniscient estimator, we compute the instantaneous performance score as

$$1 - \frac{1}{2}\left\|\pi_k^j - \pi_k^*\right\|_1, \tag{47}$$

which ensures that scores are within the $[0, 1]$ interval, where 1 connotes the best performance and 0 the worst. We tally the results for each DSE variant. Since even in a fully disconnected network, agents have access to their own observations, the lowest score is seldom zero. To account for the specific effects of network degradation (rather than observability of the HMM itself), we then re-normalize the results to $[0, 1]$ interval. In the end, we plot the average normalized performance vs. probability of link failure. The diagram that results gives insight into the robustness of the DSE method with respect to network failure

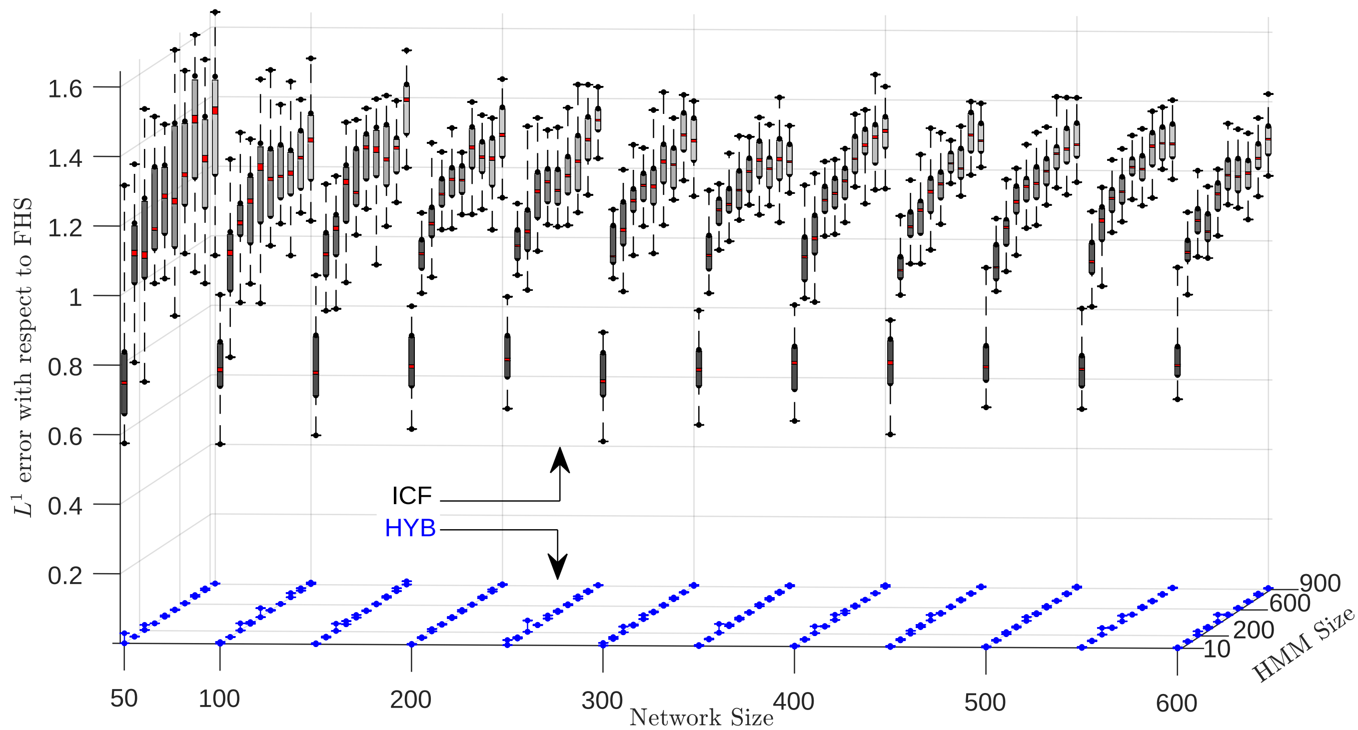


Fig. 7: Box plots of the L^1 error of the HYB and ICF methods with respect to FHS. Each box summarizes a distribution representing error over time and averaged over nodes in the network.

and gives a clear visualization of the gap between FHS and other methods.

Diagrams, as just explained, were constructed for the distributed tracking example in VI-B and another system consisting of 20 agents observing a randomly generated HMM with three states. The results can be seen in Figures 6a and 6b respectively. Some observations of interest can be made.

Figure 6a shows that p_{tr}, the threshold on probability of link failure beyond which centralized estimation is not possible, is small ($p_{tr} = 0.12$) for the target tracking example. Comparing Figure 6a to Figure 6b, the performance gap between ICF and HYB is smaller, the gap between ICF and FHS is narrower too, and the decline in performance is sharper. This can be explained by the large number of states in the HMM of the tracking example (559 states) along with the fact that there are only six observers that track the maneuvering target. The result in Figure 6a suggests that the benefit of using Hybrid method over ICF is less pronounced for systems with less accurate observation models, or on time-varying networks consisting of many small-sized, short-lived connected components.

Figure 6b illustrates a case where there are more observers, $n = 20$, and the improvement over ICF is marked and the gap between HYB and FHS is negligible. Unlike the tracking problem in Figure 6a, the gap between ICF and HYB is substantial even for values $0 \ll p$. That is because for large connected networks even if some links fail, the size of the connected components and their lifetime is much longer than those in smaller networks. This is a property of network reliability and its mathematical foundations are well-studied but beyond the scope of this paper. For our purposes it suffices to say that based on the example in Fig 6b, on reliable networks, the advantage of using Hybrid over ICF is clear. The results in Figure 6b also show that if the ratio of observers to states is large, Hybrid method performance approaches FHS even for when the probability of link failure is substantial.

Taking both examples in this section together, adopting the Hybrid method over ICF is always beneficial. Also, the improvement over ICF and the degree to which the gap with FHS is closed depends on the intrinsic properties of the HMM and underlying network.

D. Error and Communication Cost

Two additional experiments involving extensive simulations were conducted to test the comparative performance of the Hybrid method across different network sizes, across a range of HMM sizes (in terms of number of states), and along a spectrum of link failure probabilities. In the first experiment we fixed the probability of link failure to a single value ($p = 0.6$) and compared the performance of HYB with ICF over a combination of HMM and network sizes. For the second experiment, the number of states in the HMM was fixed to 10 and we measured the maximum number of consensus steps as the number of nodes in the network and link failure probabilities were varied. The first experiment aims to provide insight into the scalability of the Hybrid method, while the second experiment reports the communication cost of the method. We tried our best, in both experiments, to evaluate using random models for the network topology and the HMM that are germane to performance in realistic settings. However, some care is needed—the degree to which the quantities shown are representative will depend on the application of interest

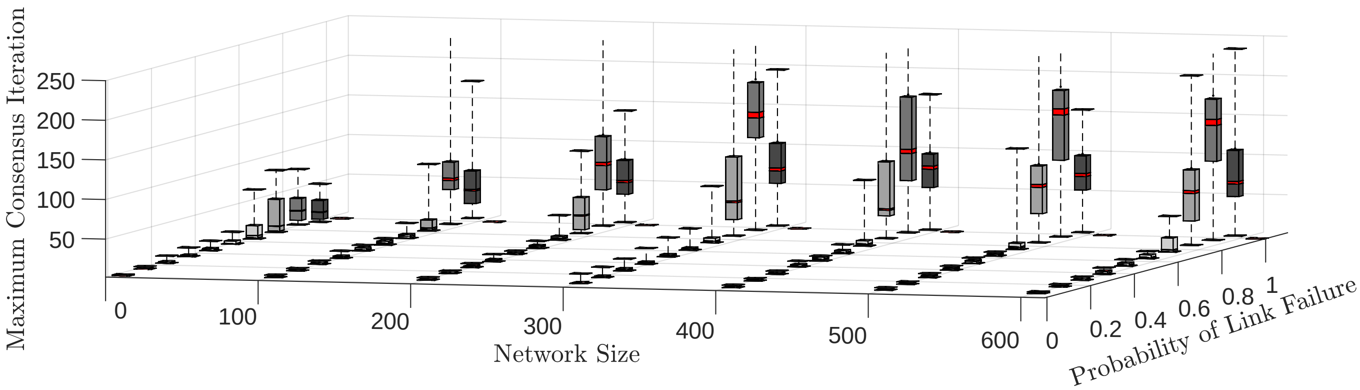


Fig. 8: The maximum number of consensus steps, each taking δt time, to reach consensus. The number of nodes and probability of link failure affects the topology, which influences the time to reach consensus.

and the communication graph underlying the system. While the numbers are only meaningful for applications with settings very similar to ours, one factor that is clearly common to all the experiments is that the Hybrid method's performance eclipses ICF in terms of L^1 error. This is further evidence for the claims established above via theoretical arguments when the assumptions of our method hold.

1) Scalability: In the first experiment, we contrast the distribution of the L^1 error of both the HYB and ICF methods with respect to FHS, across a range of network sizes while also varying the model's size (in terms of number of states). We used the generic HMM and random network model of Sections VI-A and VI-C. The parameters were set as follows: we form a grid by changing the number of observers (nodes of the random graph) from 50 to 600 with a stride size of 50, and the number of states of the underlying HMM from 10 to 900 with a stride of 100. Every point on the grid represents a pair of network nodes and HMM dimension. Then, for each pair, we simulate the estimation based on HYB, ICF, and FHS methods over 50 steps. When all the simulations have completed, we calculate the L^1 error of HYB and ICF with respect to FHS for all the nodes in the network for each time step. Taking an average of the L^1 error over the nodes produces a single number for that time step. The sequence of these average numbers over time can be seen as a distribution. This distribution is depicted as a box and whisker plot for each point of the grid in Figure 7.

It is clear from Figure 7 that, while the ICF method may incur large errors, the errors of the Hybrid method are negligible being almost zero for the whole grid. This testifies to the superiority of HYB over ICF.

Another observation is that the average error and its variance grows with the increase in the number of HMM states for a fixed network size. This can be explained by the fact that, as the number of states grow, the number of variables contributing to the error calculation also increases. A similar trend may be seen with a variance that shrinks as we move along the network size axis for a fixed HMM size. This is due to the fact that the number of nodes appears in the denominator when the average is computed, causing the variance to decrease with increasing network size.

2) Communication Cost: In this experiment we fixed the size of a random HMM to 10 states and considered different networks. For each network size with a given probability of link failure we ran the simulation for 50 steps, recording the maximum number of consensus iterations (or δt steps) across all nodes for each time-step of the simulation. This forms a measure of communication cost. Note that some nodes may converge faster than others and precisely how this occurs will depend on the network topology—the rate of convergence in a connected component is a function of the diameter of the underlying communication graph. This is quite apart from other factors, like the system model and quality of observations which do also influence the behavior of the consensus process.

We used Cauchy's convergence test with a threshold of 10^{-4} to determine when convergence had occurred. Similar to the previous experiment, we obtain a distribution for each grid point. The results are plotted in Figure 8 wherein we show the distribution of the maximum number of consensus iterations. As expected, for lower probabilities of link failure, fewer consensus steps are needed as the network has smaller graph diameter, it having connected components with more nodes. In contrast, when probability of link failure approaches 1, the components are minuscule and it takes few steps for the consensus process to converge. There is an interval over which the probability of having components with large graph diameter increases and this explains the increased consensus steps during such intervals. The jump in the number of consensus steps is much more pronounced for networks of larger size. This experiment shows that the favorable performance of the Hybrid method, illustrated in the previous experiment, is achieved in a reasonable number of consensus steps.

VII. Conclusion and Future Work

This paper has proposed a distributed state estimator for discrete-state dynamic systems with non-Gaussian noise in networks with changing topology and those that do not remain connected all the time. The method is able to achieve robustness and recover performance after an interval of disconnection. Separating the process of consensus for the correlated and uncorrelated information was the key to achieving better performance compared to ICF alone. The theoretical analysis guarantees that the method proposed in this paper has desirable convergence properties and outperforms the competitors. In many cases, this is by a significant margin. Evaluating the proposed method in a series of experiments showed considerable performance improvement compared to the state of the art in practice. The experiments also validated the mathematical analysis, showing exponential convergence under L_1 very clearly.

For future work, it would be interesting to examine the effect that topology of the network has on the performance and the gaps between ICF, HYB, and FHS. Another direction is to investigate a mixture of raw information sharing in small sub-groups of agents and conducting consensus on sub-groups.

Acknowledgements

This work was supported by the National Science Foundation in part by IIS-1302393, IIS-1453652, and ECCS-1637889.

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*, 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] F. Boem, L. Sabattini, and C. Secchi, "Decentralized state estimation for heterogeneous multi-agent systems," in *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2015, pp. 4121–4126.
[4] N. R. Ahmed, S. J. Julier, J. R. Schoenberg, and M. E. Campbell, "Decentralized bayesian fusion in networks with non-gaussian uncertainties," *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, p. 383, 2015.
[5] 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, June 2009.
[6] A. Simonetto, T. Keviczky, and R. Babuška, "Distributed nonlinear estimation for robot localization using weighted consensus," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 3026–3031.
[7] 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*, 2005, p. 9.
[8] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proceedings of IEEE Decision and Control, and European Control Conference (CDC-ECC)*, 2005, pp. 8179–8184.
[9] 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.
[10] F. Boem, R. M. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis for continuous-time nonlinear systems: The input–output case," *Annual Reviews in Control*, vol. 37, no. 1, pp. 163–169, 2013.
[11] J. Hu, D. Chen, and J. Du, "State estimation for a class of discrete nonlinear systems with randomly occurring uncertainties and distributed sensor delays," *International Journal of General Systems*, vol. 43, no. 3-4, pp. 387–401, 2014.
[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)*, 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, Jan 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] 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.
[17] G. Battistelli and L. Chisci, "Stability of consensus extended kalman filter for distributed state estimation," *Automatica*, vol. 68, pp. 169–178, 2016.
[18] D. Casbeer and R. Beard, "Distributed information filtering using consensus filters," in *Proceedings of IEEE American Control Conference (ACC)*, June 2009, pp. 1882–1887.
[19] Y. Niu and L. Sheng, "Distributed consensus-based unscented kalman filtering with missing measurements," in *Proceedings of Chinese Control Conference (CCC)*, 2017, pp. 8993–8998.
[20] L. Mao and D. W. Yang, "Distributed information fusion particle filter," in *Proceedings of International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, Aug 2014, pp. 194–197.
[21] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 3415–3422.
[22] 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.
[23] 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.
[24] 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.
[25] R. Lucchese, D. Varagnolo, J.-C. Delvenne, and J. M. Hendrickx, "Network cardinality estimation using max consensus: the case of bernoulli trials," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*. IEEE Communications Society, 2015, pp. 895–901.
[26] H. Terelius, D. Varagnolo, and K. H. Johansson, "Distributed size estimation of dynamic anonymous networks," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*. IEEE conference proceedings, 2012, pp. 5221–5227.
[27] R. Lucchese and D. Varagnolo, "Networks cardinality estimation using order statistics," in *Proceeedings of IEEE American Control Conference (ACC)*, 2015, pp. 3810–3817.
[28] A. Tamjidi, R. Oftadeh, S. Chakravorty, and D. Shell, "Efficient distributed state estimation of hidden markov models over unreliable networks," in *Proceedings of IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2017, pp. 112–119.
[29] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-gaussian dependence," in *Proceedings of International Conference on Information Fusion (FUSION)*, 2012, pp. 1876–1883.
[30] J. Ajgl and M. Šimandl, "Design of a robust fusion of probability densities," in *Proceeedings of IEEE American Control Conference (ACC)*, 2015, pp. 4204–4209.
[31] E. Seneta, *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006.
[32] B. D. Anderson, "Forgetting properties for hidden markov models," in *Proceedings of US/Australia Joint Workshop on Defence Applications of Signal Processing*. Amsterdam: Elsevier, 2001, pp. 26–39.
[33] C. Liverani, B. Saussol, and S. Vaienti, "A probabilistic approach to intermittency," *Ergodic theory and dynamical systems*, vol. 19, no. 3, pp. 671–685, 1999.