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*—A new recursive consensus filter for distributed state estimation on Hidden Markov Models (HMMs) is presented and shown to be well-suited to multi-robot settings and associated applications. 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 assumed. The algorithm is a Hybrid method: Iterative Conservative Fusion (ICF) is used to reach consensus over potentially correlated priors, while consensus over likelihoods 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 Hybrid method converges exponentially to the ideal distributed estimator, despite the latter being conceptual and utterly unworkable in practice. The method is extensively evaluated in a series of simulated experiments and its performance surpasses competing algorithms.**

I. Introduction

Mobile and robotic-sensor networks have many applications and the problem of estimation within such networks has, thus, been a topic of extensive study in recent years [?], [?], [?]. In a robotic-sensor network, robots carry sensors that make noisy observations of the state of an underlying system of interest. Their estimation process is considered centralized if all the nodes send their raw observations to a central node that is responsible for calculating an estimate based on the collective information [?]. This is not always possible owing to link failures as well as bandwidth and energy constraints [?].

One alternative —which we term 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. The Channel Filter [?], a classic method in this category, 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 fully recover the performance of the centralized estimator so long as the network is fully connected and time invariant. A similar bookkeeping-based approach, but which relaxes the directed communication graph requirement, was proposed by Bahr, Walter and Leonard [?]. 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.

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 [?].

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

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

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 [?], [?]. 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,

[1]Department of Aerospace Engineering, Texas A&M University
[2]Department of Computer Science & Engineering, Texas A&M University
Corresponding author email: ahtamjidi@tamu.edu

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 agent's perform Iterative Conservative Fusion (ICF) on node posteriors [?], [?], [?]. 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 prior owing to early termination of a consensus process, [?] and [?] proposed to use a combination of CL and ICF. To justify their method they refer to the complimentary 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, the proposed combination can bring about the complementary benefits of both of them.

The precedence of [?] and [?] in their amalgamation of CL and ICF must be acknowledged openly: the present authors were working on the method described herein, albeit with the distinct motive of developing approaches to operate under conditions of severe network degradation, and encountered their work only after our first publication on the subject [?]. In essence, our assumptions on the network are different and, consequently, the analysis and final results are different from [?], [?]. (In detail Proposition 1 yields the quality of network necessary to achieve some desired performance—a question that is only meaningful when the network is not assumed to be fully connected, as was assumed in [?], [?].) 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. However, handling priors with ICF and new information with CL reaps the benefits and, as we will show, the performance in settings where the communication network is unreliable is outstanding—so good in fact as to eclipse previously envisioned domains of applicability.

This article is an extended version of the conference paper [?], where method of [?] was generalized to finite-state systems with non-Gaussian noise. We define and examine a *Hybrid* method that is a synergistic combination of CL and ICF for Hidden Markov Models. Compared to [?], in this paper we have added mathematical analysis and proof for superiority of our method over ICF. We also show through discussion 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. Thus, the method should be the first choice in many applications.

Figure 1 shows different estimation methods and illustrates how the proposed Hybrid method can be situated. The horizontal axis in the diagram is the probability of link failure: $p = 0$ means that when 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 measure, yet to be defined, that quantifies the similarity of the estimated PMF 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 DSE under connections of intermittent connectivity.

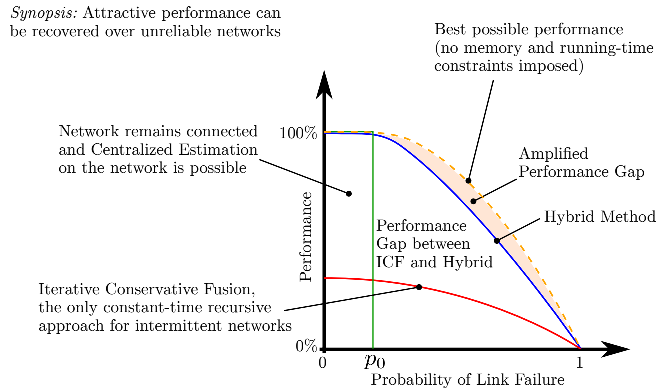


Fig. 1: The synopsis of the paper shown graphically. A centralized method can produce the highest quality estimates possible, but can only be used when the network remains fully connected for all time. In contrast, Iterative Conservative Fusion (a constant time recursive approach) is practicable with intermittent connectivity, but its conservative nature means that information is diluted, degrading estimation performance. The method described herein 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.

Note that in the event of intermittent network disconnection, the estimator performance will inevitably fall 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 band on the proximity measure. We show that our method's performance approaches the upper bound, resulting in a large performance improvement compared with ICF.

The remainder of the paper is organized as follows. In Section II, the notation used in this paper is explained. The section also identifies the assumptions and describes the system model. Section III 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 IV and, finally, we evaluate its performance in Section V.

II. 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 $\overline{\mathcal{N}}_i$ represent neighbors of node v_i that are connected by an edge to v_i. The set $\mathcal{N}_i = \overline{\mathcal{N}}_i \cup \{v_i\}$ 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:

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

Robot i is said to be part of a component CC^m if $v_i \in \text{CC}^m$, where the m 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 robot 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 robot 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 robot i is connected (via some path) to robot 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 and observation made by agent i at step k, respectively. The realizations of those random variables at step k are denoted $\mathbf{x}_k$ and $\mathbf{z}_k^i$.
- The transition model is an $n_s \times n_s$ matrix written $\mathcal{P}_{k|k-1} \triangleq \text{p}(\mathbf{X}_k | \mathbf{X}_{k-1})$. All the agents possess this model.
- Each agent has an observation model, which is an $n_s \times n_z$ matrix written as $\text{p}(\mathbf{Z}_k^i | \mathbf{X}_k), i \in \boldsymbol{I}_n$. The observation models of different agents may differ.
- The prior, prediction, and posterior probabilities are $1 \times n_s$ random vectors

$$\pi_{k-1} \triangleq \text{p}\left(\mathbf{X}_{k-1} | \{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_{k-1}}^{i \in \boldsymbol{I}_n}\right),$$
$$\tilde{\pi}_k \triangleq \text{p}\left(\mathbf{X}_k | \{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_{k-1}}^{i \in \boldsymbol{I}_n}, \mathbf{X}_{k-1}\right),$$
$$\pi_k \triangleq \text{p}\left(\mathbf{X}_k | \{\mathbf{z}_k^i\}_{k \in \boldsymbol{I}_k}^{i \in \boldsymbol{I}_n}\right),$$

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:

$$\mathbf{X}_{k+1} = f(\mathbf{X}_{k+1}, \mathbf{w}_k) \quad \mathbf{w}_k \sim \text{p}(\mathbf{W}_k), \tag{2}$$
$$\mathbf{Z}_{k+1}^i = h^i(\mathbf{X}_{k+1}, \mathbf{v}_k) \quad \mathbf{v}_k \sim \text{p}(\mathbf{V}_k), \tag{3}$$

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 the communication channel which 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 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 $\mathbf{R}_k^{ij}, j \in \overline{\mathcal{N}}_i$ denotes the information that node i receives from node j, its neighbor at time 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 normalizing constant.)

III. 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

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

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 insight that in (4) one can see that if all agents share the same prior information, they will recover the centralized estimator's performance if 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 probabilities (log-likelihood), that is,

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

Once consensus is reached, 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 use m to indicate consensus iterations throughout this paper. On a communication graph G one can use a message passing protocol of the form

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

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

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

With this messaging passing protocol

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

Note that for each node i, the γ_{ij}'s depend only 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. This means that if the consensus process converges, Introducing the concise notation

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

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 node priors being equal. Next, we discuss the conventional method employed for cases with node priors that are not equal.

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 which 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 straightforwardly thereafter.

Conservative approximation of a PMF is only possible under certain conditions. Bailey, Julier, and Agamennoni [**?**] 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:

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

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, $0 \leq \omega \leq 1$. 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 $\tilde{\pi}^i$. The GMD fusion of these estimates is also a conservative approximation of the centralized estimate.

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

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

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

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

Remark 4. It is shown in [**?**] that raising a PMF to the 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

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

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 $j \in \text{CC}_k^i$, the local variables $\phi^j(m)$ converge to a unique ϕ^*. Moreover, ϕ^* is a convex combination of initial consensus variables of all the agents in the set CC_k^i, that is, for all $i \in \text{CC}_k^i$,

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

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

To repeat the process iteratively, set $\pi_{k+1}^j = \phi^*, \forall j \in \text{CC}_k^i$ 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

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

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

$${}^{\text{ICF}}\pi_k = \tau^k\left(\pi_0\right) = \tau\left(\tau\left(\cdots\tau\left(\pi_0\right)\cdots\right)\right). \tag{26}$$

This shows that under general conditions, even with the same initial PMF, the ICF method will not generate the same estimate as CE or CL do over time. 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 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 $\pi_0 = {}^{\text{ICF}}\bar{\pi}_0$. The striking benefit is that we can 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.

Input: π_{k-1}^i

1 Use (6) to calculate $\tilde{\pi}_k^i$

2 Collect local observation z_k^i and calculate $\mathcal{O}_k^i$ and $\tilde{l}_k^i$

3 Initialize consensus variables:

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

4 $m = 0$

5 **while** *not converged* **do**

6   BROADCAST$[\psi^i(m), \phi^i(m)]$

7   RECEIVE$[\psi^j(m), \phi^j(m)] \quad \forall j \in \mathcal{N}^i$

8   Collect received data

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

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

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

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

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

11     $m \leftarrow m + 1$

12 **end**

13 Calculate posteriors according to:

$$ \pi_k^i = e^{|\text{CC}_k^i| \psi^i(m)} \phi^i(m). $$

Algorithm 1: Hybrid Method

IV. 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. Our method is presented in detail as pseudo-code in Algorithm 1.

To give a meaningful interpretation, imagine a scenario consisting of n agents observing $\mathbf{x}_k$, the state of a Markov chain at time k, and communicating with each other through a time-varying network 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 2 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 13 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.

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. If one wishes to analyze the performance of an estimator by comparing its efficiency to an ideal estimator, this poses a dilemma. 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)* (outlined below) incorporates all the information possible while respecting network topology constraints and, thus, constitutes the proper upper limit for estimator performance.

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 IV.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 II. 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 estimator 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} and 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 as ${}^{\text{HYB}}\pi_{\text{CC}^m,k_0}$. 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 [?], 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 [?], [?]. Moreover, one can use the results of [?] 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 is clearly visible in the plots showing the method's empirical performance, as 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 IV.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 $\mathrm{CC}^j_{t_0}, \mathrm{CC}^j_{t_1}, \dots \mathrm{CC}^j_{t_m}$. 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, $t \in \{t_0, t_1, \dots, t_m\}$ 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} robot belongs to a connected component whose members are fixed. Then, from Lemma IV.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 $N_{\epsilon_1} = \max_k(\log_{\rho_{t_k}} u - \log_{\rho_{t_k}} r_{t_k})$. 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 $t_{c,k} = t_{k+1} - t_k$. We introduce a constant that bounds the estimation operation as follows. 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 robots. Then we have

$$\sum_{t=t_0}^{t0+T}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1 = \sum_k\sum_{t=t_k}^{t_{k+1}}\left\|{}^{\text{FHS}}\pi^j_t - {}^{\text{HYB}}\pi^j_t\right\|_1. \tag{43}$$

Which can be further expanded into

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

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 can arrive at N_{ϵ_2} as calculated in (35). □

V. Experiments

We conduct an analysis of the comparative performance of our method in two ways. First, we examine two case studies (see V-A and V-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.

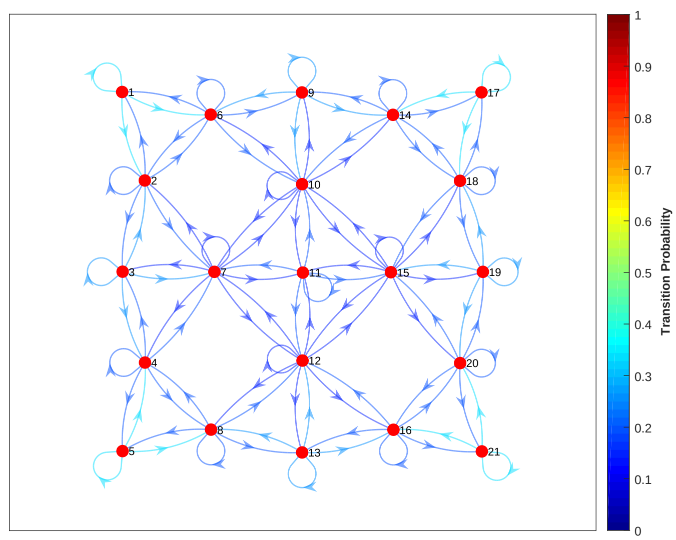


Fig. 2: A schematic of the model used in case study V-A. The Markov Model has 21 states, and is being observed by five agents over an unreliable network.

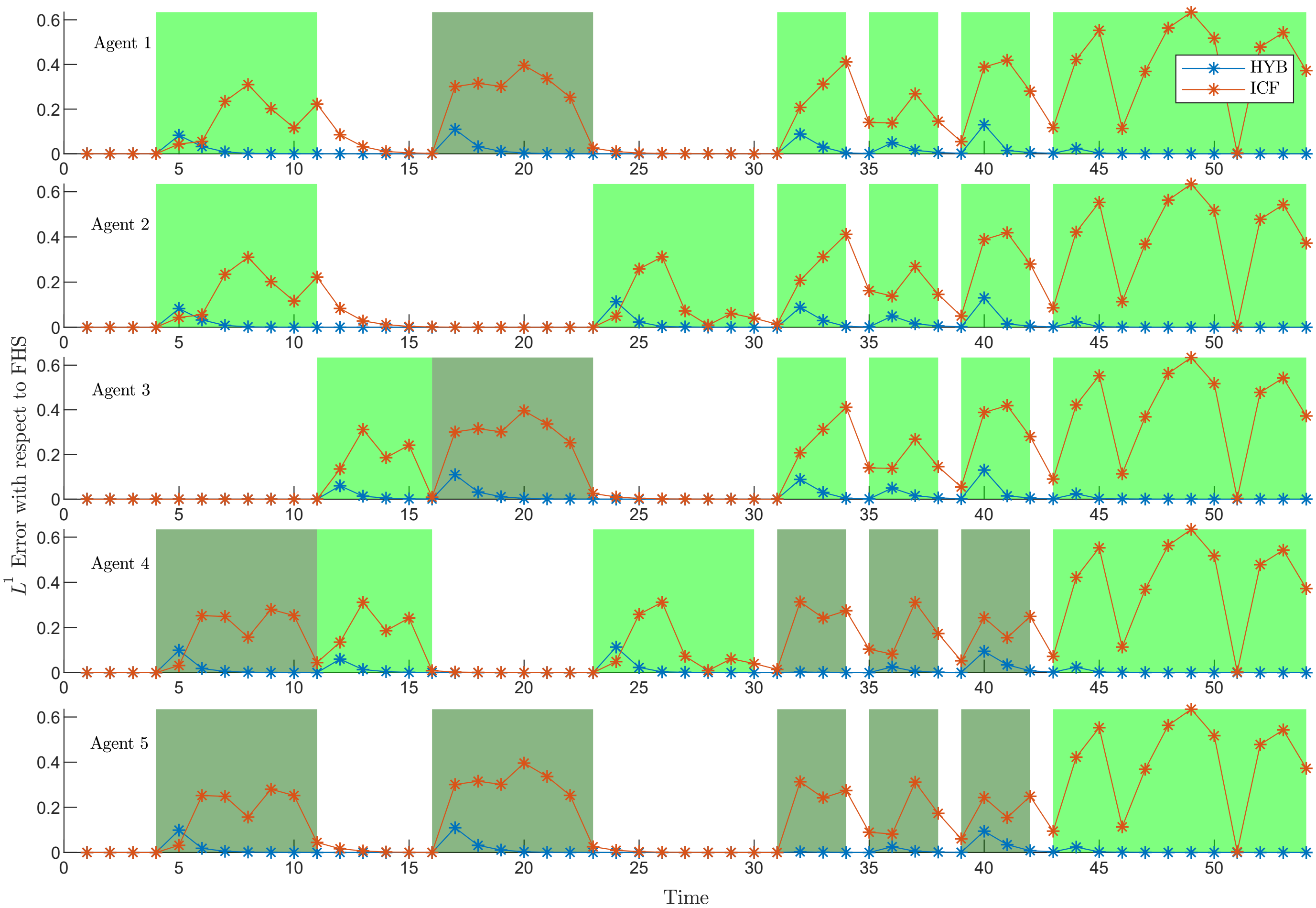


Fig. 3: Performance comparison of HYB and ICF on the Markov Model in Figure 2. The shaded areas mark the lifetime of components and agents with the same shade of color belong to the same component.

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 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 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

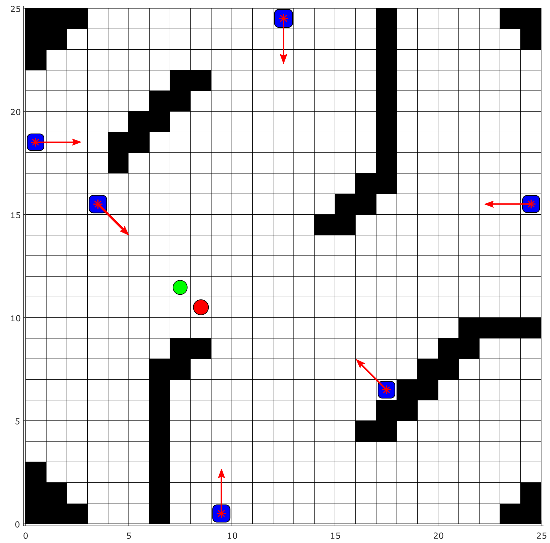


Fig. 4: The grid-based map of the environment for the tracking example V-B. The dark cells depict obstacles; blue circles are trackers and the red circle is the ground truth location of the maneuvering target; the green circle depicts an observation made by an agent.

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 a 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 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—separate agents each performing its own Bayesian update independently. However, while the Hybrid method is able to start to recover immediately after reconnection, ICF continues with degraded performance even after reconnection owing to the fact that it ignores the correlations.

C. Controlled 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 method and FHS. The experiments reported in this subsection were performed is as follows.

We take the HMM and construct a fully connected communication network. 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 $[0, 1]$ 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 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 V-B and another system consisting of 20 agents observing a 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 the threshold beyond which centralized estimation is not possible ($p_0 = 0.12$) is small 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 drastic 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

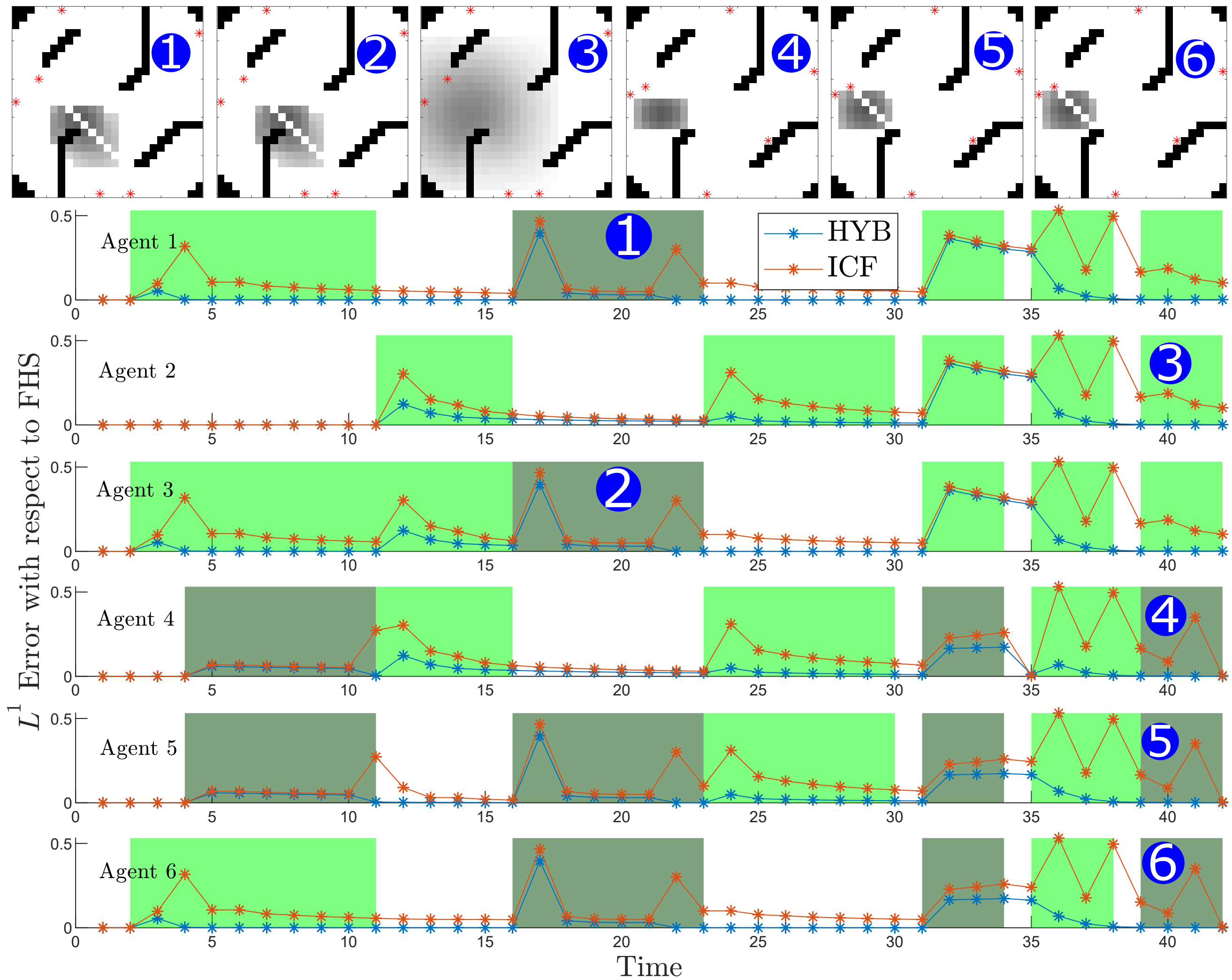


Fig. 5: Estimation performance for the tracking case study (see Figure 4 for the environment and details of the scenario). For each agent its distribution at a single step is shown at the top and is indexed accordingly at its representative time.

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.

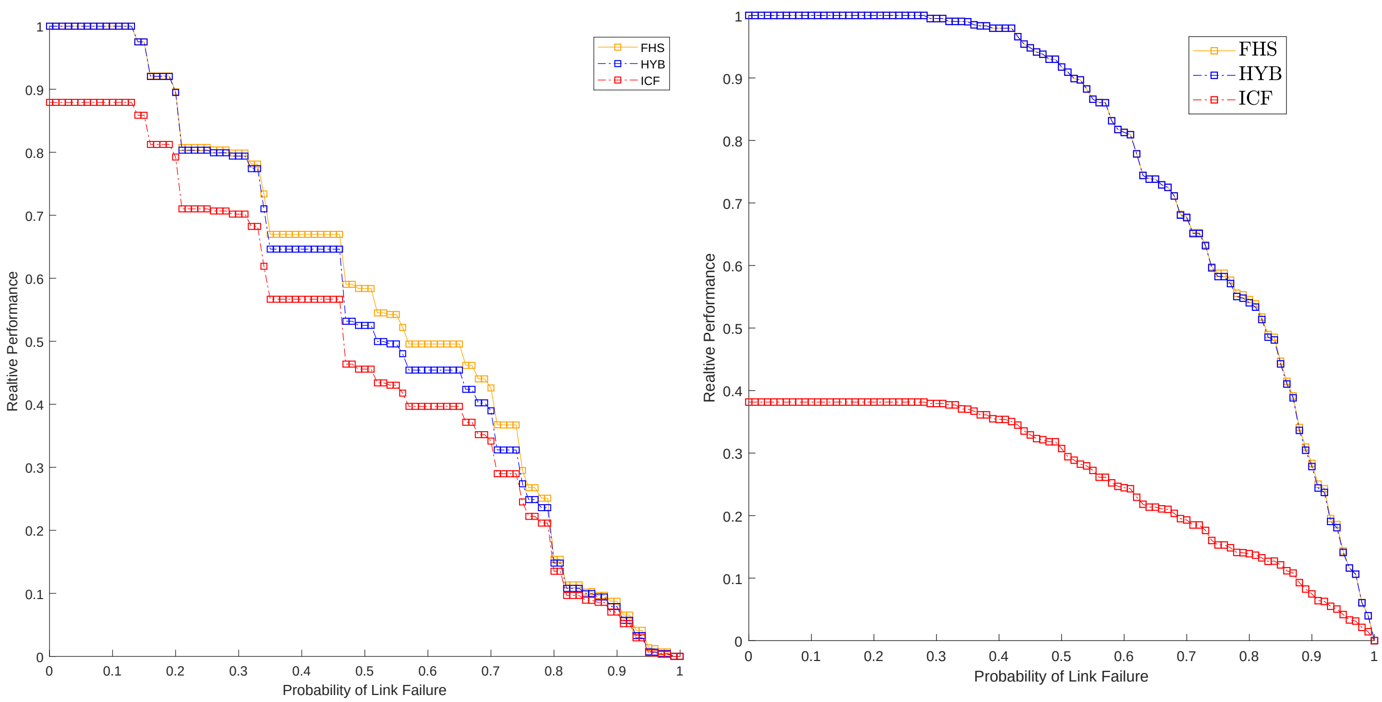
Taking both examples in this subsection together, adopting the Hybrid method over ICF is always beneficial. Also, the improvement over ICF and the degree to which the gap with FHS is closed depends on the intrinsic properties of the HMM and underlying network.

VI. Conclusion

This paper proposes 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. Theoretical analysis guarantees that the method proposed in this paper will show promising 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.

Acknowledgements

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



(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 probability of link failure, moving from left to right represents 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.