Efficient Distributed State Estimation of Hidden Markov Models over Unreliable Networks

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

***Abstract*—This paper presents a new recursive Hybrid consensus filter for distributed state estimation on a Hidden Markov Model (HMM), which is well suited to multi-robot applications and settings. The proposed algorithm is *scalable*, *robust to network failure* and capable of handling *non-Gaussian* transition and observation models and is, therefore, quite general. No global knowledge of the communication network is assumed. 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). The proposed method is evaluated in a multi-agent tracking problem and a high-dimensional HMM and it is shown that its performance surpasses the competing algorithms.**

I. Introduction

Estimation within robotic-sensor networks has many applications and, thus, has been extensively studied in recent years [1], [2], [3]. In a mobile 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 who then calculates an estimate based on the collective information [4]. This is not always possible owing to link failures as well as bandwidth and energy constraints [5]. One viable alternative is to distribute the process.

In Distributed State Estimation (DSE), the processor on each robot (or node) fuses local information with the incoming information from neighbors and redistributes the fused result on the network. The objective is to design both a protocol for message passing between nodes and local fusion rules so that the nodes reach a consensus over their collective information. Although the DSE algorithms are not guaranteed to match the performance of the centralized estimator all the time, their scalability, modularity and, robustness to network failure motivates the ongoing research. These features are important for the envisioned robotic applications of such algorithms, such as multi-agent localization [6] and cooperative target tracking [7].

DSE algorithms can be categorized on the basis of assumptions they make. Any DSE method makes assumptions about one or more of the following aspects: the state (static [8] or dynamic [6]), state transition model (linear [9] or non-linear [10], [11], [12], [13]), type of noise (Gaussian [8], [9] or non-Gaussian [14]), topology of the network (constant or changing [15], [8]), connectivity of the network (always [10] or intermittent connection [15], [8]), agents' knowledge about the network topology (global or local [15], [8], [10]) and finally the treatment of mutual information between local estimate (exact solution through bookkeeping [1] or conservative solutions that avoid double counting [16], [17]).

The research on DSE for linear systems with Gaussian noise is extensive (see [9], [18] for reviews). For nonlinear systems with Gausssian noise, the distributed versions of Extended Kalman Filters (EKF), Extended Information Filters (EIF) and Unscented Kalman Filter (UKF) have been proposed [19], [20], [10], respectively. For nonlinear systems with non-Gaussian noise, different flavors of Distributed Particle Filter (DPF) methods were proposed by Mao and Yang [21].

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 perform consensus on likelihoods only [22], [23]. We refer to this approach as Consensus on Likelihoods (CL). The advantage of CL is that given enough time to reach consensus, it can match the centralized estimator's performance. However, if the network becomes disconnected, priors start to differ between nodes, and CL methods fail. In such scenarios, the prevailing approach is to perform Iterative Conservative Fusion (ICF) on node posteriors [24], [16], [17]. 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.

Researchers recently began combining ICF and CL methods to benefit from their complementary features [19], [10], [15]: CL can reach consensus over uncorrelated new information, while ICF can handle correlated prior information. In this paper we extend our previous method [15] to general finite-state systems with non-Gaussian noise. We propose a *Hybrid* method that is a synergistic combination of CL and ICF. The contributions of the paper are:

- The new Hybrid method that shows unmistakable

[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

superiority over ICF in applications with large number of agents and time-varying networks that face intermittent disconnection.
- The method handles non-Gaussian noise models, being particularly useful for collaborative tracking and localization applications.

Hidden Markov Models (HMMs) are adopted as the general means for describing the system whose state is being estimated. In Section II, the notation as well as the assumptions used in this paper are explained, along with the system model. Section III provides some preliminaries on distributed state estimation, paving the way for the new method. Our proposed method is presented in Section IV and, finally, we evaluate its performance in Section V.

II. Notation and Modeling

The Network Topology: Assume that we have n homogeneous agents associated with the nodes of a graph. These agents can communicate with each other under a time-varying undirected network topology $G_k = \langle \mathcal{V}, \mathcal{E}_k \rangle$ where $\mathcal{V}$ and $\mathcal{E}_k$ are the set of graph nodes and edges at step k respectively. The node corresponding to the i^{th} agent is denoted by v_i. If $(v_i, v_j) \in \mathcal{E}_k$, it means that agents i and j can communicate directly at step 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. For an arbitrary set with $s \in \mathbb{N}$ 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. We 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.

The Model: Consider a finite state HMM with the following specification:

- 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 as $\mathbf{x}_k$ and $\mathbf{z}_k^i$.
- The transition model is a $n_s \times n_s$ matrix denoted as $\mathcal{P}_{k|k-1} \triangleq \mathrm{p}(\mathbf{X}_k | \mathbf{X}_{k-1})$. All the agents possess this model.
- Each agent has an observation model, which is a $n_s \times n_z$ matrix denoted as $\mathrm{p}(\mathbf{Z}_k^i | \mathbf{X}_k), i \in \boldsymbol{I}_n$. The observation models of different agents can 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 well-defined 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:

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

in which, $\mathbf{W}_k$ and $\mathbf{V}_k$ are random variables representing dynamics and observation noise. We further 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. Note that the above specification for the HMM and the model may easily 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. Moreover, 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 proposed method 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. We proceed to define a method, within the Consensus on Likelihoods (CL) class, called *Distributed Consensus Based Filtering* which is particular to systems where agents have identical prior information. Given that network disconnection and early stopping of the consensus process results in different priors among the agents, we review *Conservative Fusion* and its iterative version as a remedy for such cases. After reading this section, the reader should find it straightforward to understand the logic behind the proposed method as outlined in the subsequent section.

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 assumptions, has the form

$$\mathrm{p}(\mathbf{X}_k|\mathbf{z}_k) = \frac{1}{\eta}\mathrm{p}(\mathbf{z}_k|\mathbf{X}_k)\mathrm{p}(\mathbf{X}_k|\mathbf{z}_{k-1},\mathbf{X}_{k-1}) \tag{3}$$
$$= \frac{1}{\eta}\prod_{i=1}^{n}\mathrm{p}(\mathbf{z}_k^i|\mathbf{X}_k)\int \mathrm{p}(\mathbf{X}_k|\mathbf{X}_{k-1})\mathrm{p}(\mathbf{X}_{k-1}|\mathbf{z}_{k-1})\mathrm{d}\mathbf{X_{k-1}}.$$

Performing recursive estimation in a sensor network setting and for an HMM can be done in one of the following ways:

Centralized Estimation 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

$$\tilde{\pi}_k = \pi_{k-1}\mathcal{P}_{k|k-1}, \tag{4}$$

then, updating via

$$\pi_k = \frac{1}{\eta}\tilde{\pi}_k\mathcal{O}_k, \tag{5}$$

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

Distributed Consensus Based Filtering is based on the insight that in (3) one can see that if all agents share the same prior information, they can recover the centralized estimator's performance if they can reach a 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),i.e.,

$$\tilde{l}_k = \frac{1}{n}\log\prod_{i=1}^{n}\mathcal{O}_k^i = \frac{1}{n}\sum_{i=1}^{n}\log\mathcal{O}_k^i = \frac{1}{n}\sum_{i=1}^{n}\tilde{l}_k^i. \tag{6}$$

Once consensus is reached, the updated estimate is

$$\pi_k = \frac{1}{\eta}\overbrace{\underbrace{\pi_{k-1}}_{\text{prior}}\mathcal{P}_{k|k-1}}^{\text{prediction}}\underbrace{e^{n\tilde{l}_k}}_{\text{likelihood}}. \tag{7}$$

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

$$\psi^i(m+1) = \textstyle\sum_{j=1}^{|\mathcal{N}_i|}\gamma_{ij}(m)\psi^j(m), \tag{8}$$
$$\text{s.t. } \sum_m \gamma_{ij}(m) = 1,\ \psi^i(0) = \tilde{l}_k^i,$$

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, and

$$\gamma_{ij}(m) = \begin{cases} \frac{1}{1+\max\{d_i(m),d_j(m)\}} & \text{if } (i,j)\in\mathcal{E}_m, \\ 1 - \sum\limits_{(i,n)\in\mathcal{E}}\gamma_{in} & \text{if } i=j, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

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 can be recovered. The prerequisite for this method to work is that the network remains connected. This requirement, however, is too restrictive for many applications. *Conservative Approximate Distributed Filtering* is an approach where, instead of putting effort into keeping the dependencies between agents' information, a fusion rule is designed to guarantee that no double counting of mutual information occurs. This usually results in the replacement of independent information with some form of conservative approximation. Such a treatment results in inferior performance, compared to the exact distributed filter's output that is degraded.

Since *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. The mechanics of conservative fusion becomes easier to understand thereafter.

Conservative approximation of a Probability Mass Function (PMF) is possible under certain conditions. The authors in [25] introduced a set of sufficient conditions for a PMF $\tilde{\mathrm{p}}(\mathbf{X})$ to satisfy in order to be a conservative approximation of $\mathrm{p}(\mathbf{X})$, a second PMF. The conditions are

- the property of non-decreasing entropy:
$$\mathrm{H}(\mathrm{p}(\mathbf{X})) \leq \mathrm{H}(\tilde{\mathrm{p}}(\mathbf{X})),$$
- the order preservation property so that,
$$\mathrm{p}(\mathbf{x}_i) \leq \mathrm{p}(\mathbf{x}_j) \text{ iff } \tilde{\mathrm{p}}(\mathbf{x}_i) \leq \tilde{\mathrm{p}}(\mathbf{x}_j),\ \forall \mathbf{x}_i,\mathbf{x}_j \in \mathbf{X}.$$

Conservative Fusion of two PMFs (CF) can be achieved for two probability distribution functions

$p_a(\mathbf{X}|\mathbf{I_a})$ and $p_b(\mathbf{X}|\mathbf{I}_b)$, with the *Geometric Mean Density Rule (GMD)*:

$$\begin{aligned}p_c(\mathbf{X}) &= \frac{1}{\eta_c} p_a(\mathbf{X}|\mathbf{I_a})^{\omega} p_b(\mathbf{X}|\mathbf{I_b})^{1-\omega} \\ &= \frac{1}{\eta_c} p_a(\mathbf{X}|\mathbf{I_a} \setminus \mathbf{I}_b)^{\omega} p_b(\mathbf{X}|\mathbf{I_b} \setminus \mathbf{I}_b)^{1-\omega} p_a(\mathbf{X}|\mathbf{I_a} \cap \mathbf{I}_b),\end{aligned} \tag{10}$$

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.

$$\tilde{\pi}_k = \frac{1}{\eta} \prod_{i=1}^{n} (\tilde{\pi}_k^i)^{\omega_i}, \quad \text{s.t.} \quad \textstyle\sum_{i=1}^{n} \omega_i = 1. \tag{11}$$

Remark 1. Several criteria have been proposed to choose the ω_i. One such criterion is [26]:

$$\tilde{\pi} = \arg\min_{\pi} \max_{i} \{\mathcal{D}(\pi \| \tilde{\pi}^i)\}, \tag{12}$$

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

Remark 2. In [25], it is shown that raising a PMF to the power of $\omega \leq 1$ reduces its entropy. From (11) it can be seen that applying the GMD rule reduces the entropy of the likelihood probabilities that are independent. This is undesirable and can be avoided by treating the likelihood probabilities separately.

Iterative CF (ICF) is achieved as follows. At 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

$$\begin{aligned}&\omega^* = \arg\min_{\omega} \mathcal{J}\big(\frac{1}{\eta} \textstyle\prod_{j \in \mathcal{N}^i(m)} \big[\phi^j(m)\big]^{\omega_j}\big), \\ &s.t. \quad \textstyle\prod_{j \in \mathcal{N}^i(m)} \omega_j = 1, \quad \forall j, \quad \omega_j \geq 0,\end{aligned} \tag{13}$$

where η is the normalization constant and $\mathcal{J}(\cdot)$ is an optimization objective function. Specifically, it can be entropy $H(\cdot)$ or the criteria in (12). The ϕ^is are then updated locally for the next iteration as

$$\phi^i(m+1) = \frac{1}{\eta^*} \textstyle\prod_{j \in \mathcal{N}^i(m)} \big[\phi^j(m)\big]^{\omega_j^*}. \tag{14}$$

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

$$\lim_{m \to \infty} \phi^i(m) = \phi^* = \frac{1}{\eta} \textstyle\prod_{j \in \boldsymbol{I}_{\text{CC}_k^i}} \big[\phi^j(0)\big]^{\omega_j^*}, \forall i \in \text{CC}_k^i. \tag{15}$$

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 $k+1$.

IV. Hybrid ICF and CL

We propose a Hybrid approach that uses ICF to reach consensus over priors and the Distributed Consensus Based Filtering method (outlined above, which we refer to as CL hereafter) for distributed averaging of local information updates. Our method is summarized in Algorithm 1. Imagine a scenario consisting of n agents, observing $\mathbf{x}_k$, the state of a Markov chain at time k, that are 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 agents calculate their own local prediction $\{\tilde{\pi}_k^i\}^{i \in \boldsymbol{I}_n}$ (line 1 in the algorithm). Then they 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 find a consensus over the priors using (13) recursively. The CL approach is used to form the consensus over the new information available to agent i from other agents 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 group 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 and passes them to its neighbors.

To better understand the benefit of the Hybrid method, we compare the estimator performances under centralized, ICF, and Hybrid schemes. The following set of equations compares the estimates

$$^{\text{CEN}}\pi_{k+1} = \frac{1}{\eta}\ ^{\text{CEN}}\tilde{\pi}_k \prod_{j \in \boldsymbol{I}_n} \mathcal{O}_k^j, \tag{16}$$

$$^{\text{ICF}}\pi_{k+1}^i = \frac{1}{\eta'} \prod_{j \in \boldsymbol{I}_{\text{CC}_k^i}} \left[^{\text{ICF}}\tilde{\pi}_k^j\right]^{\omega_j^{\text{ICF}}} \prod_{j \in \boldsymbol{I}_{\text{CC}_k^i}} \left[\mathcal{O}_k^j\right]^{\omega_j^{\text{ICF}}}, \tag{17}$$

$$^{\text{HYB}}\pi_{k+1}^i = \frac{1}{\eta''} \prod_{j \in \boldsymbol{I}_{\text{CC}_k^i}} \left[^{\text{HYB}}\tilde{\pi}_k^j\right]^{\omega_j^{\text{HYB}}} \prod_{j \in \boldsymbol{I}_{\text{CC}_k^i}} \mathcal{O}_k^j. \tag{18}$$

Remark 3. The value produced as a prediction from the centralized scheme is approximated by both the ICF and Hybrid methods. Hybrid's performance gain over ICF comes from the fact that, unlike ICF, it does not down-weight observations. This improvement becomes increasingly pronounced as number of agents increase, showing the *scalibility* of the Hybrid method.

Remark 4. The above equations demonstrate why, unlike ICF, the Hybrid method is capable of recovering the centralized estimate if, after a disconnection, the network remains path-connected long enough thereafter. Assume that a subset of agents resume connection at time k after having being disconnected for some time. The agents will have different priors at time k, and thus their update steps are computed with either (17) for ICF, or (18) for Hybrid. In general, at this point, either of the posteriors can be closer to the centralized estimate. However, as time goes forward, only the Hybrid method will recover the centralized estimate if the system retains path connectivity globally. This is because the priors of the centralized estimate and the Hybrid method must reconcile due to the forgetting property of Markov chains [27]. According to this property a recursive estimator on an HMM will exponentially forget the initial distribution of the HMM. This implies that, after an extended interval of path connectivity, the mismatch of local priors becomes irrelevant and the Hybrid method can reap the benefit of its consensus-based fusion part. So doing, it will match the performance of the centralized estimator. In contrast, ICF will continue fusing the local estimates with a conservative fusion rule, incurring a performance loss when compared to the centralized and Hybrid estimators.

Remark 5. The line of reasoning in the preceding remark holds for connected components as well. This provides evidence for the effectiveness of the Hybrid estimator's performance modulo network connectivity, and thus its *robustness* to network failure.

Algorithm 1: Hybrid Method

Input : π_{k-1}^i

Use (5) to calculate $\tilde{\pi}_k^i$

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

Initialize consensus variables

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

$m = 0$

while *NOT CONVERGED* **do**

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

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

    Collect received data

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

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

$$\phi^i(m+1) = \texttt{ICF}(\mathcal{C}^i(m))$$

    Do one iteration of MHMC on consensus variables for new information

$$\psi^i(m+1) = \texttt{MHMC}(\mathcal{M}^i(m))$$

    $m = m + 1$

Calculate the posteriors according to:

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

V. Experiments

The first experiment is concerned with a decentralized target pose estimation problem in a grid using multiple observers connected through a changing topology network. Fig. 1 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 deterministic; 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, normal to its direction of motion, as shown in the figure. The beam hits either the target or an obstacle. In the former 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; in the latter case, under the assumption of no false positives, the observer produces a 'no target' symbol as 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).

At each 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 has two components; one has the rook observers and the other one has the bishops. The observers in each component

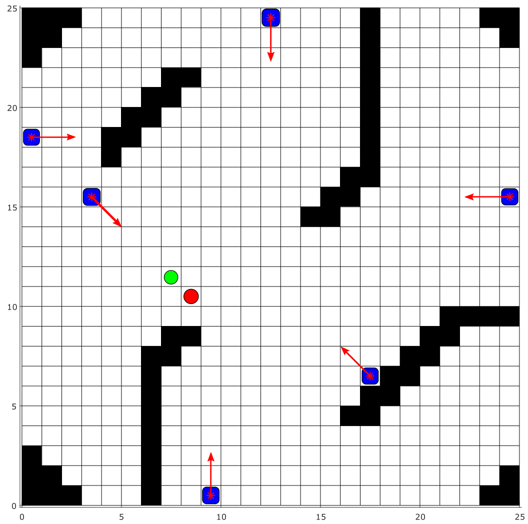

Fig. 1. The grid map of the environment, dark cells depict obstacles; blue circles are trackers and the red circle is the ground truth location of the maneuvering target; the green circle depicts the observation of an agent.

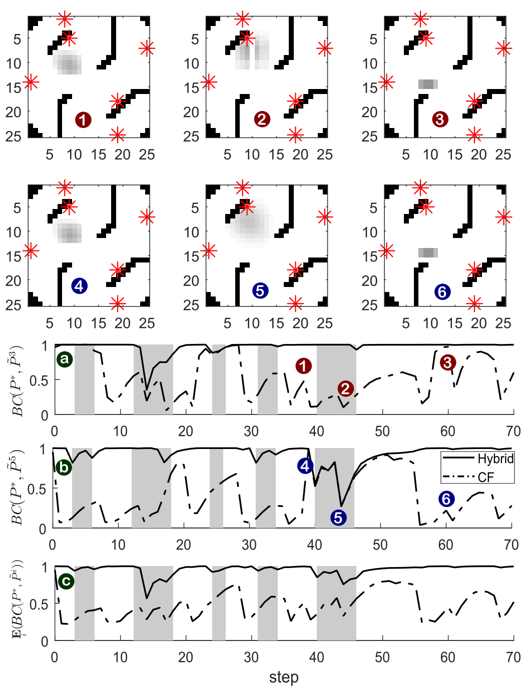

Fig. 2. Estimation performance in the tracking example

are always connected while the link between the two components is intermittent. All communications occur at a higher rate than Markov transition steps, which allows the connected nodes to reach consensus over the shared information.

We evaluate the performance of the proposed Hybrid method during the phase where the rooks become disconnected from bishops, and are then reconnected after some interval. For purposes of comparison, each node 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. The third instance concerns a hypothetical god's-eye-view centralized estimator, to give a baseline for comparison. To quantify differences, we use the Bhattacharyya Coefficient [28] between the estimation results and the centralized estimator. The Bhattacharyya coefficient can be used to evaluate the similarity of two probability mass functions, $\pi_1(\mathbf{X}), \pi_2(\mathbf{X})$ as:

$$BC(\pi_1(\mathbf{X}), \pi_2(\mathbf{X})) = \sum_{\mathbf{x}\in\mathbf{X}} \sqrt{\pi_1(\mathbf{x})\pi_2(\mathbf{x})}. \quad (19)$$

In the case of complete similarity, $p_1 = p_2$, we have $BC(p_1, p_2) = 1$. Moreover, $BC(p_1, p_2) = 0$ describes maximal dissimilarity.

Fig. 2 compares the performance of the Hybrid and ICF methods, showing that the proposed method outperforms CF and is able to recover performance very close to centralized solution after reconnection. Based on the Bhattacharyya coefficient, closeness between centralized and decentralized estimates drops during the interval of network partition. This is expected, since observers do not have access to all the information available to the centralized estimator. 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.

Fig. 2 also gives a detailed view of the performance of the Hybrid method and compares the estimation results of observer 3, a rook, and observer 5, a bishop, during three different time steps. The shaded area shows the time during which rooks are disconnected from the bishops. The higher difference between centralized and decentralized estimate for the fifth observer can be explained based on the fact that the bishop has less information at its disposal. However, after reconnection both groups are able to converge to the same value, which is very close to the centralized estimator.

In a second experiment we have evaluated the robustness of the proposed method for networks with different likelihoods of link failure. We report the Bhattacharyya coefficient vs. link failure probability for a general decentralized HMM with a network of size 20 and state size 30 with each node roughly connected to 10% of the other nodes. We simulate the system multiple times, each time for 150 time steps but with different probability of link failure. At the beginning of each step, a 2 regular graph with 15 nodes is generated and, given a probability of failure for each link, some links in the graph will randomly be disconnected. The graph still remains connected some portion of the time, but this depends on

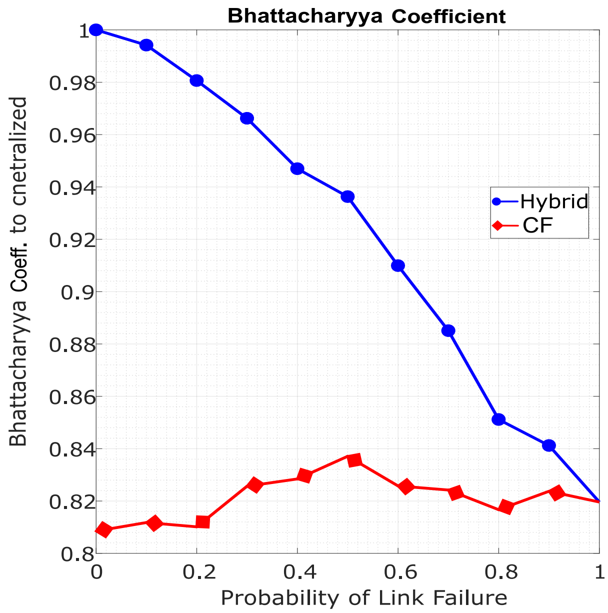

Fig. 3. Performance comparison between the proposed method and ICF.

the degree and probability of failure. If the regularity degree goes down or the probability of failure increases, more often than not, the graph becomes disconnected. In practice, for $p \geq 0.05$, consensus methods that rely on full connectivity no longer succeed since the network almost always suffers disconnection at some point in time.

We ran our method for 150 steps for a range of probabilities of link failure and compared performances with the ideal centralized result (which is obtained by assuming full connectivity at all times). The performance is evaluated by calculating the average value for the Bhattacharyya coefficient and determinant ratio measure at all steps and for all receptors. Based on Fig. 3, for the case considered in this experiment, our decentralized estimator performs close to the ideal centralized one for $p \in [0.0, 0.1]$, drastically outperforming ICF in all cases. This means that in the case considered here, our method can perform almost as well as the ideal estimator for an unreliable network. Obviously the performance can vary from one system to another and under different network topologies, but the results show that the method can recover the performance of the centralized method despite operating on an unreliable network and, moreover, substantially outperforms ICF. This latter fact has also already been established theoretically.

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. Separating the process of consensus for the correlated and uncorrelated information was the key to achieving better performance compared to ICF alone. Evaluating the proposed method on a multi-agent tracking application and a high-dimensional HMM distributed state estimator problem showed substantial performance improvement compared to the state of the art. We are able to achieve robustness and recover performance after an interval of disconnection.

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 *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 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,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2383–2395, June 2009.

[6] A. Simonetto, T. Keviczky, and R. Babuka, “Distributed nonlinear estimation for robot localization using weighted consensus,” in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 3026–3031.

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

[8] 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*. IEEE Press, 2005, p. 9.

[9] R. Olfati-Saber, “Distributed kalman filter with embedded consensus filters,” in *Decision and Control, and European Control Conference. CDC-ECC'05*. IEEE, 2005, pp. 8179–8184.

[10] 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.

[11] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, “Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations,” *IEEE transactions on automatic control*, vol. 48, no. 2, pp. 246–258, 2003.

[12] 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.

[13] 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.

[14] 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.

[15] A. Tamjidi, S. Chakravorty, and D. Shell, "Unifying consensus and covariance intersection for decentralized state estimation," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, 2016, pp. 125–130.

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

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

[18] 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.

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

[20] D. Casbeer and R. Beard, "Distributed information filtering using consensus filters," in *American Control Conference. ACC*, June 2009, pp. 1882–1887.

[21] L. Mao and D. W. Yang, "Distributed information fusion particle filter," in *Intelligent Human-Machine Systems and Cybernetics, Sixth International Conference on*, vol. 1, Aug 2014, pp. 194–197.

[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] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-gaussian dependence," in *Information Fusion (FUSION), 2012 International Conference on*, 2012, pp. 1876–1883.

[26] J. Ajgl and M. Šimandl, "Design of a robust fusion of probability densities," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 4204–4209.

[27] B. D. Anderson, "Forgetting properties for hidden markov models," in *Proc. AFOSWDSTO Workshop*, 1997.

[28] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: The Indian Journal of Statistics*, pp. 401–406, 1946.