Distributed Information Bayesian Recursive Update Filter

Lili Wang, Keith A. LeGrand, Shreyas Sundaram

***Abstract*— This paper addresses consensus-based networked estimation of the state of a nonlinear dynamical system. This paper first presents an information form of the recently proposed Bayesian recursive update filter (BRUF), a Kalman filter that uses a recursive update to incorporate information from nonlinear measurement systems. Under the assumptions that the system is collectively observable and the network is strongly connected, a distributed information Bayesian recursive update filter (DIBRUF), a distributed form of the information Bayesian recursive update filter (IBRUF), is proposed, which exploits consensus on information vectors and matrices. Compared to the distributed extended Kalman filter (DEKF), the DIBRUF reduces the linearization error of the extended Kalman filter (EKF) by dividing the measurement update into N steps. Unlike the BRUF and IBRUF, which require local observability, the DIBRUF requires only the network to be collectively observable, as the sensors can share information among the network. Simulation experiments demonstrate the validity of the proposed approach.**

I. Introduction

Nonlinear state estimation is a common problem in various fields such as robotics, aerospace, finance, and biological systems, where the underlying dynamics exhibit nonlinear behavior. The techniques developed are essential for accurately tracking and predicting the behavior of these systems in real-world scenarios.

The extended Kalman filter (EKF) is an extension of the classic Kalman filter for nonlinear systems. It linearizes the system dynamics and measurement equations around the current estimate to perform state estimation. While widely used, EKF may suffer from approximation errors, particularly in highly nonlinear systems, and is sensitive to the quality of initial estimates.

The iterated EKF [1] can improve the linearization error compared to the EKF by recursively recomputing the center of the truncated Taylor series expansion (TSE) in a process known as relinearization. A comparable but fundamentally different nonlinear update known as the recursive update filter (RUF) was later proposed in [2]. While both the iterated EKF and RUF employ recursive relinearization steps to reduce linearization error, the RUF divides the measurement update into N attenuated updates instead of the iterated EKF's single large update about the final TSE center. The same authors later proposed the Bayesian recursive update filter (BRUF) [3] [4] as a variant of the RUF derived from a Bayes-theoretic perspective.

Distributed state estimation has received increasing attention in recent years due to its broad application to sensor networks, robotics, distributed control systems, and wireless communications. Distributed state estimation involves estimating system states using information gathered from multiple spatially or functionally distributed sensors or agents.

The study of distributed state estimation for linear systems can be dated back to the so-called distributed Kalman filter problem [5], which involves system and measurement noise in the problem formulation [6], [7]. Most available Kalman filter based approaches [6]–[9] require the agents to both share "signal information", which can be measurements or local state estimates, and fuse certain "structural information", which forms the covariance or information matrix of the nominal centralized Kalman filter for each agent. Notable existing Kalman filter based approaches to the distributed state estimation for nonlinear systems include those based on the EKF [10]–[14] and unscented Kalman filter (UKF) [15].

Distributed state estimation offers several advantages over centralized approaches, including scalability, fault tolerance, among other benefits. In this paper, we want to develop a distributed state estimation algorithm considering the most recent RUF algorithms aiming to increase the estimation accuracy and decrease the communication cost. We study this by considering the problem of enabling a network of agents to estimate the state of a discrete-time nonlinear dynamical system. First, we introduce an algorithm called the information Bayesian recursive update filter (IBRUF), which is the information form of the BRUF. Based on this, an algorithm called distributed information Bayesian recursive update filter (DIBRUF) is proposed for distributed estimation. At its core, DIBRUF leverages the Bayesian framework to iteratively update state estimates based on incoming measurements and prior information. However, its distributed nature sets it apart, which allows it to handle large-scale systems where data is generated and processed across multiple interconnected nodes or agents. The integration of IBRUF with consensus algorithms from the distributed algorithm field enables distributed systems to perform better estimation of complex dynamic processes while ensuring scalability, robustness, and fault tolerance. This combination finds applications in sensor networks, multi-robot systems, distributed control, and decentralized decision-making, among others.

The paper is organized as follows. In Section II, the

Lili Wang and Keith LeGrand are with the School of Aeronautics and Astronautics at Purdue University ({wang6127, klegrand}@purdue.edu).

Shreyas Sundaram is with the Elmore Family School of Electrical and Computer Engineering at Purdue University (sundara2@purdue.edu).

problem formulation is proposed. In Section III, the information BRUF is first proposed, and then the distributed information BRUF is discussed. Section IV demonstrates the effectiveness of the proposed filters via simulations. Section V concludes the paper.

II. Problem Formulation

This paper considers the distributed state estimation over a network in which each agent can process local data and communicate data with neighbors. Consider a network of $m > 0$ (possibly mobile) autonomous agents labeled $1, 2, \ldots, m$. An agent j is said to be a *neighbor* of agent i if it can transmit information to agent i; agent i is always considered a neighbor of itself. Denote by $\mathcal{N}_i$ the set of labels of agent i's neighbors. Neighbor relations are characterized by a directed graph $\mathbb{N}$ with m vertices and a set of arcs defined so that there is an arc in $\mathbb{N}$ from vertex j to vertex i whenever agent j is a neighbor of agent i. Since each agent i is always a neighbor of itself, $\mathbb{N}$ has a self-arc at each of its vertices. At each time-step t, each agent i receives a discrete-time signal $z_i(t) \in \mathbb{R}^{s_i},\ i \in \mathbb{N}_m = \{1, 2, \ldots, m\}$, where

$$z_i(t) = h_i(x(t)) + v_i(t), \quad i \in \mathbb{N}_m \tag{1}$$

$$x(t+1) = f(x(t)) + w(t), \tag{2}$$

and $x \in \mathbb{R}^n$ represents the state of the system. The functions f and h_i, $i \in \mathbb{N}_m$, represent the nonlinear dynamics and measurement structures of the agents, respectively. It is assumed that $w(t) \sim \mathcal{N}(0, Q)$ and $v_i(t) \sim \mathcal{N}(0, R_i)$, $i \in \mathbb{N}_m$. The process noise and measurement noise are assumed to be uncorrelated, and their respective covariance matrices Q and R_i are assumed to be positive definite.

The objective is to have, at each time $t \in \{1, 2, \ldots\}$ and for each node $i \in \mathbb{N}_m$, an estimate $x_i(t)$ of the state $x(t)$ constructed based on only the local measurements $z_i(t)$ and on the data received from its neighbors $j \in \mathcal{N}_i$.

III. Main results

In this section, a new information form of the BRUF called the IBRUF is proposed. Based on this new information form, the DIBRUF is then introduced.

A. The Information Bayesian Recursive Update Filter

Before describing the proposed DIBRUF algorithm, it is convenient to briefly introduce the centralized IBRUF, which assumes that the measurements

$$z(t) = h(x(t)) + v(t)$$

are aggregrated and processed by a single central processor. Here, $h(x(t))$ and $v(t)$ are the stacked vector of $h_i(x(t))$ and the stacked vector of $v_i(t)$ for $i \in \mathbb{N}_m$, respectively.

As shown in [3], [4], for BRUF the measurement Jacobian is recomputed at the current state estimate for each update step in the BRUF update. This results in a better representation of the Bayesian posterior compared to a single EKF update, which is computed in a single step using a linearization about the prior state estimate. It can be observed that the measurement Jacobian is computed using the current state estimate for N steps, where the tunable parameter N represents the number of recursions.

Instead of updating the estimated states, the information vector has the advantage of simplifying the calculation. Thus, the IBRUF is proposed in Algorithm 1, wherein the *information* matrices $\Lambda^\imath \triangleq (P^\imath)^{-1}$ and the *information* vectors $q^\imath = \Lambda^\imath x^\imath$ are be propagated for $\imath = 1, 2, \ldots, N$.

Algorithm 1 generalizes the IBRUF update, corresponding to $z(t) = H^i x(t)$ and $x(t+1) = Ax(t)$ via the EKF paradigm of linearization of the measurement and the state equations around the current estimate. With this respect, the following assumption is needed.

Assumption 1: The functions $h_i : \mathbb{R}^n \to \mathbb{R}^{s_i}$, $i \in \mathbb{N}_m$ and $f : \mathbb{R}^n \to \mathbb{R}^n$ are twice continuously differentiable on $\mathbb{R}^n$, where $n = \dim(x)$.

Note that the information vector $q^\imath$ is updated based on the innovation term $z - h(x^{\imath-1}) + H^\imath x^{\imath-1}$, and the information matrix Λ is computed using the inflated measurement noise $\frac{1}{c_\imath} R$. By recursively recalculating the measurement Jacobian, the IBRUF update provides a better approximation of the state estimate than a single EKF update. For the linear case, the performance of Algorithm 1 is equivalent to the performance of a single Kalman update with measurement covariance R for linear measurements.

Algorithm 1 The Information Bayesian Recursive Update Filter

Require: $\bar{x}$, the prior state estimate; $\bar{P}$, the prior covariance; z, a measurement; R, the measurement covariance; N, the number of steps; $\bar{\Lambda} = \bar{P}^{-1}$; $\bar{q} = \bar{\Lambda}\bar{x}$; $\sum_{\imath=1}^{N} c_\imath = 1$ where $c_\imath > 0$

Correction:

Initialize $q^0 = \bar{q},\ \Lambda^0 = \bar{\Lambda},\ x^0 = \bar{\Lambda}^{-1}\bar{q}$

for $\imath = 1, 2, \ldots N$ **do**

$H^\imath = \frac{\partial h(x)}{\partial x}|_{x=x^{\imath-1}}$

$q^\imath = q^{\imath-1} + c_\imath (H^\imath)^T R^{-1}(z - h(x^{\imath-1}) + H^\imath x^{\imath-1})$

$\Lambda^\imath = \Lambda^{\imath-1} + c_\imath (H^\imath)^T R^{-1} H^\imath$

$x^\imath = \Lambda^\imath q^\imath$

end for

Output: $\hat{q} = q^N,\ \hat{\Lambda} = \Lambda^N$

Prediction:

$\hat{x} = (\hat{\Lambda})^{-1}\hat{q}$

$A = \frac{\partial f(x)}{\partial x}|_{x=\hat{x}}$

Let $W = Q^{-1}$

$\Lambda^+ = W - WA(\hat{\Lambda} + A^T W A)^{-1} A^T W$

$\hat{x}^+ = f(\hat{x})$

Output: $\bar{x} = \hat{x}^+,\ \bar{\Lambda} = \Lambda^+$

Theorem 1: Given prior information matrix $\bar{\Lambda}$, the information vector $\bar{q}$ and linear measurement $z = Hx$, if $\sum_{\imath=1}^{N} c_\imath = 1$ where $c_\imath > 0$ the following can be shown.

(1) The Nth iterative information covariance update

$$\Lambda^{\imath} = \Lambda^{\imath-1} + c_{\imath} H^T R^{-1} H, \quad \imath = 1, \ldots, N \tag{3}$$

is equivalent to the information filter update

$$\hat{\Lambda} = \bar{\Lambda} + H^T (R)^{-1} H \tag{4}$$

where $\Lambda^0 = \bar{\Lambda}$. That is, $\Lambda^N = \hat{\Lambda}$.

(2) The Nth iterative information vector update

$$q^{\imath} = q^{\imath-1} + c_{\imath} H^T R^{-1} z, \quad \imath = 1, \ldots, N \tag{5}$$

is equivalent to the information filter update

$$\hat{q} = \bar{q} + H^T R^{-1} z \tag{6}$$

where $q^0 = \bar{q}$. That is $q^N = \hat{q}$.

The proof is omitted here due to page length limitation.

It is worth stating the relationship between the Nth update and the standard Kalman update. The following result can be derived.

Theorem 2: Given prior covariance matrix $\bar{P}$, prior information matrix $\bar{\Lambda} \triangleq \bar{P}^{-1}$, the prior state estimate $\bar{x}$, the information vector $\bar{q} \triangleq \bar{\Lambda}\bar{x}$ and linear measurement $z = Hx$, the following can be shown.

(1) The Nth iterative information covariance update

$$\Lambda^N = \hat{P}^{-1} \tag{7}$$

where $\hat{P}$ is the Kalman update defined as $\hat{P} = (I - KH)\bar{P}$. Here I is an identity matrix with an appropriate size, and $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$ is the gain matrix.

(2) The Nth iterative information vector update

$$q^N = \Lambda^N \hat{x}$$

where $\hat{x}$ is the Kalman update defined as $\hat{x} = \bar{x} + K(z - H\bar{x})$. Here K is the Kalman gain $K = \bar{P}H^T(H\bar{P}H^T + R)^{-1}$.

The proof is omitted here due to page length limitation.

B. The Distributed Information Bayesian Recursive Update Filter

The focus of this paper is on the development of DIBRUF. First, the concept of observability of nonlinear systems is discussed. Define a map and a vector as follows:

$$\phi_i(z) \triangleq \begin{bmatrix} h_i(z)^T & h_i(f(z))^T & \ldots & h_i(f^M(z))^T \end{bmatrix}^T \tag{8}$$

where f^M is the function that f composed with itself M times, ϕ_i is a map such that $\phi_i : \mathbb{R}^n \to \mathbb{R}^{ns_i}$, $z \in \mathbb{R}^n$. Let $\phi(\cdot)$ be the stacked map of ϕ_is, that is

$$\phi = \begin{bmatrix} \phi_1^T & \ldots & \phi_m^T \end{bmatrix}^T : \mathbb{R}^n \to \mathbb{R}^{\sum_{i=1}^m ns_i}.$$

In other words, given a time window $\{t - M, \ldots, t\}$, ϕ coincides with the mapping from the state x at time $t - M$ to the vector made up of the noise-free collective outputs at times $t - M, \ldots, t$.

According to the complete observability definition in [16], the corresponding definition of observability on a network of agents is defined below.

Definition 1: The system (1), (2) is *collectively observable* if there is a positive integer M such that the invertibility of the mapping $\phi(x)$ holds for each t.

According to [11], in order to ensure the observability of the system (1), (2) the following assumptions are made. Note that Assumption 2 ensures that $f(x)$ is a diffeomorphism on $\mathbb{R}^n$. Assumption 3 ensures that the system is collectively observable as $\phi(x)$ is invertible.

Assumption 2: For any $x \in \mathbb{R}^n$, $\partial f(x)/\partial x$ is nonsingular.

Assumption 3: There exists a positive number M such that, for any $x \in \mathbb{R}^n$, $\text{rank}\,\{\partial\phi(x)/\partial x\} = n$.

Assumption 4: The neighbor graph is strongly connected.

A graph is considered strongly connected if and only if there is a path from any vertex in the graph to any other vertex. In other words, the information obtained by each agent can be shared among the network.

Consensus is a widely exploited tool for distributed computation over a network, including optimization, averaging, and so on. The basic idea of consensus is to collectively reach an agreement over the whole network by iteration, where each node does computation based on information only from its neighbors. In this paper, consensus is applied to fuse the information provided by each agent. Two local information pairs are considered. The pair $\left(\Lambda_i^N, q_i^N\right)$ represents the prior information, and $\left(\delta\Lambda_i^N, \delta q_i^N\right)$ represents the novel information. Performing consensus on information matrices [10] is more efficient than an equivalent consensus on covariance matrices as it avoids costly matrix inversions.

At each time t (and where hereon t is omitted for brevity), let $\delta q_i^N(0) = \delta q_i^N$, $\delta\Lambda_i^N(0) = \delta\Lambda_i^N$, $q_i^N(0) = \bar{q}_i^{N-1}$, and $\Lambda_i^N(0) = \Lambda_i^{N-1}$ after the IBRUF's N recursions for each agent i. Each agent updates its estimate with a given number L consensus steps for $l = 1, 2, \ldots, L$:

$$\begin{aligned}
\Lambda_i^N(l) &= \sum_{j \in \mathcal{N}_i} a_{ij} \Lambda_j^N(l-1) \\
q_i^N(l) &= \sum_{j \in \mathcal{N}_i} a_{ij} q_j^N(l-1) \\
\delta\Lambda_i^N(l) &= \sum_{j \in \mathcal{N}_i} a_{ij} \delta\Lambda_j^N(l-1) \\
\delta q_i^N(l) &= \sum_{j \in \mathcal{N}_i} a_{ij} \delta q_j^N(l-1)
\end{aligned}$$

Here, $a_{ij} > 0$ if $j \in \mathcal{N}_i$, and $a_{ij} = 0$ otherwise. It is assumed that $\sum_{i \in \mathcal{N}_i} a_{ij} = 1$. Thus, in each consensus iteration, each node i computes a local average as a convex combination of the values of its neighbors with suitable consensus weights a_{ij}s.

The DIBRUF algorithm is summarized in Algorithm 2. To analyze the proposed algorithm's stability, some conditions on the nonlinearity of the system are first established. According to Algorithm 2, functions f and h_i can be represented in the following forms:

$$f(x) - f(\hat{x}_i) = A_i(x - \hat{x}_i) + \xi_i(x, \hat{x}_i)$$
$$h_i(x) - h_i(x_i^{\imath-1}) = H_i(x - x_i^{\imath-1}) + \Xi_i^{\imath}(x, x_i^{\imath-1})$$

where $\xi(\cdot)$ and $\Xi^\imath(\cdot)$ are suitable continuous functions going to zero as $x-\hat{x}_i$ goes to zero, and $x-x_i^{\imath-1}$ goes to zero respectively.

Assumption 5: There exist positive numbers ϵ_ξ, κ_ξ, ϵ_{Ξ_i}, and κ_{Ξ_i}, such that the nonlinear functions ξ and Ξ are bounded as

$$\|\xi_i(x,\hat{x})\| \leq \kappa_{\xi_i}\|x-\hat{x}\|^2$$

$$\|\Xi_i^\imath(x,\hat{x})\| \leq \kappa_{\Xi_i}\|x-\hat{x}\|^2$$

for any pairs $x, \hat{x} \in \mathbb{R}^n$ satisfying $\|x-\hat{x}\| \leq \epsilon_\xi$, and $\|x-\hat{x}\| \leq \epsilon_{\Xi_i}$, respectively.

The following result can be derived according to [11]. Due to page length limitation, the proof is omitted here.

Theorem 3: Let the system trajectory belong to a compact set $\mathcal{X}$. Under Assumptions 1-5, suppose that DIBRUF is initialized at time $t=1$ with a positive definite matrix $\bar{P}_i$. Then, the estimation error $e_i(t)=x(t)-\hat{x}_i(t)$ is bounded for any $i \in \mathbb{N}_m$. That is, $\|e_i(t)\| \leq \epsilon$ provided that the initial error $e_i(0)$, process noise $w(t)$, and measurement noise $v_i(t)$ are bounded.

IV. Simulation

A target tracking case is provided to illustrate the performance of the DIBRUF. The target motion is modeled by a linear model $x(t+1)=Ax(t)+w(t)$ with

$$A=\begin{bmatrix}1 & T_s & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & T_s\\ 0 & 0 & 0 & 1\end{bmatrix},\quad Q=\begin{bmatrix}\frac{T_s^3}{3} & \frac{T_s^2}{2} & 0 & 0\\ \frac{T_s^2}{2} & T_s & 0 & 0\\ 0 & 0 & \frac{T_s^3}{3} & \frac{T_s^2}{2}\\ 0 & 0 & \frac{T_s^2}{2} & T_s\end{bmatrix} q$$

where $x(t)=\begin{bmatrix}x_x(t) & \dot{x}_x(t) & x_y(t) & \dot{x}_y(t)\end{bmatrix}^T$ is the target state at time t with $(x_x(t), x_y(t))$ being the position of the target, and $(\dot{x}_x(t), \dot{x}_y(t))$ being the velocity of the target respectively; Q is the covariance matrix of the disturbance $w(t)$; q is the variance of the random fluctuations of the target speed; and T_s is the sampling time interval.

The target is measured by a group of sensors with angle or distance measurement capability. That is, for each sensor $i \in \mathbb{N}_m$,

$$h_i(t)=\begin{cases}h_i^a & \text{if } i \text{ is a direction sensor}\\ h_i^r & \text{if } i \text{ is a distance sensor}\\ \begin{bmatrix}h_i^a\\ h_i^r\end{bmatrix} & \text{if } i \text{ is a sensor for both measurements}\end{cases} \tag{9}$$

where $h_i^a = \arctan(x_y - s_y^i, x_x - s_x^i)$, and $h_i^r = \sqrt{(x_x-s_x^i)^2+(x_y-s_y^i)^2}$. Here (s_x^i, s_y^i) is sensor i's position. The measurement noise is assumed to have $\sigma_\theta = 5^o$ standard deviation for angle measurements and $\sigma_r = 20m$ standard deviation for distance measurements. In this simulation, we choose $T_s = 1s$, and $q = 0.5m^2/s^3$. We consider a network of six sensors which are positioned as shown in Fig. 1. The blue trajectory is the true trajectory of the target with an initial state $[500m, -1m/s, 0, 0]^T$. Each sensor has both angle and distance measurement capabilities, but the sensing abilities are not always active.

Algorithm 2 The Distributed Information Bayesian Recursive Update based Algorithm

For $i \in \mathbf{m}$

Require: $\bar{x}_i$, the prior state estimate; $\bar{P}_i$, the prior covariance; z_i, a measurement; R_i, the measurement covariance; N, the number of steps; L, the number of consensus updates; $\bar{\Lambda}_i = \bar{P}_i^{-1}$; $\bar{q}_i = \bar{\Lambda}_i \bar{x}_i$

Correction:

Initialize $q_i^0 = \bar{q}_i$, $\Lambda_i^0 = \bar{\Lambda}_i$, $x_i^0 = \bar{\Lambda}_i^{-1}\bar{q}_i$

for $\imath = 1, 2, \ldots N$ **do**

$H_i^\imath = \frac{\partial h(x)}{\partial x}|_{x_i=x_i^{\imath-1}}$

$\delta q_i^\imath = (H_i^\imath)^T (R_i)^{-1}(z_i - h_i(x_i^{\imath-1}) + H_i^\imath x_i^{\imath-1})$

$\delta\Lambda_i^\imath = (H_i^\imath)^T (R_i)^{-1} H_i^\imath$

$q_i^\imath = q_i^{\imath-1} + c_\imath \delta q_i^\imath$

$\Lambda_i^\imath = \Lambda_i^{\imath-1} + c_\imath \delta\Lambda_i^\imath$

$x_i^\imath = \Lambda_i^\imath q_i^\imath$

end for

Consensus:

Require: $\delta q_i^N(0) = \delta q_i^N$, $\delta\Lambda_i^N(0) = \delta\Lambda_i^N$, $q_i^N(0) = \bar{q}_i^{N-1}$, $\Lambda_i^N(0) = \Lambda_i^{N-1}$

for $l = 1, \ldots, L$ **do**

$\delta q_i^N(l) = \sum_{j\in\mathcal{N}_i} a_{ij}\delta q_j^N(l-1)$

$\delta\Lambda_i^N(l) = \sum_{j\in\mathcal{N}_i} a_{ij}\delta\Lambda_j^N(l-1)$

$q_i^N(l) = \sum_{j\in\mathcal{N}_i} a_{ij} q_j^N(l-1)$

$\Lambda_i^N(l) = \sum_{j\in\mathcal{N}_i} a_{ij}\Lambda_j^N(l-1)$

end for

$\hat{q}_i = q_i^N(L) + \delta q_i^N(L)$

$\hat{\Lambda}_i = \Lambda_i^N(L) + \delta\Lambda_i^N(L)$

Output: $\hat{z}_i$, $\hat{\Lambda}_i$

Prediction:

$\hat{x}_i = (\hat{\Lambda}_i)^{-1}\hat{z}_i$

$A_i = \frac{\partial f(x)}{\partial x}|_{x=\hat{x}_i}$

Let $W = Q^{-1}$

$\Lambda_i^+ = W - WA_i(\hat{\Lambda}_i + A_i^T W A_i)^{-1} A_i^T W$

$\hat{x}_i^+ = f(\hat{x}_i)$

Output: $\bar{x}_i = \hat{x}_i^+$, $\bar{\Lambda}_i = \Lambda_i^+$

Three scenarios are considered to illustrate the filters proposed in this paper. Scenario 1 shows the performance of IBRUF. Scenario 2 shows the performance of DIBRUF. Scenario 3 demonstrates the necessity of collective observability. In each simulation, each c_i is chosen to be $\frac{2i}{N(N+1)}$. It can be checked that $\sum_{i=1}^N c_i = 1$.

Scenario 1: Performance of IBRUF

First, consider the case when only sensor 1 is active with both measurements. That is, $h(t) = \begin{bmatrix}h_1^a(t), h_1^r(t)\end{bmatrix}^T$.

Two different values of N, which are 1 and 10, are chosen to show the effectiveness of the IBRUF (Algorithm 1). Note that when $N=1$, the IBRUF becomes an EKF. The trajectories of the estimates are plotted in Fig. 2.

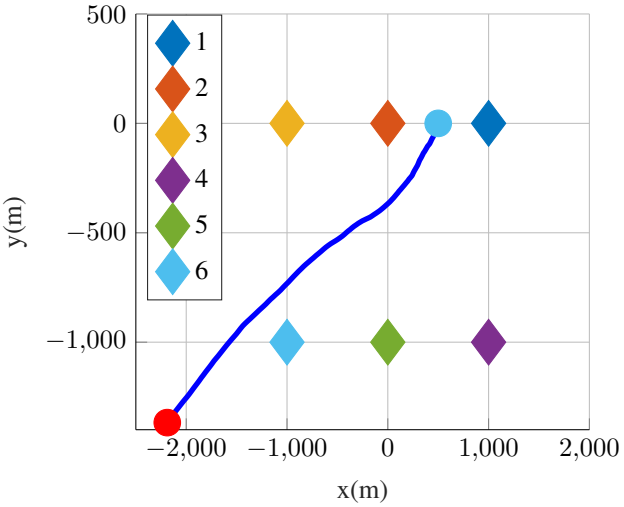

Fig. 1: Sensor locations and target trajectory, where the cyan dot marks the initial target position and the red dot marks the terminal target position.

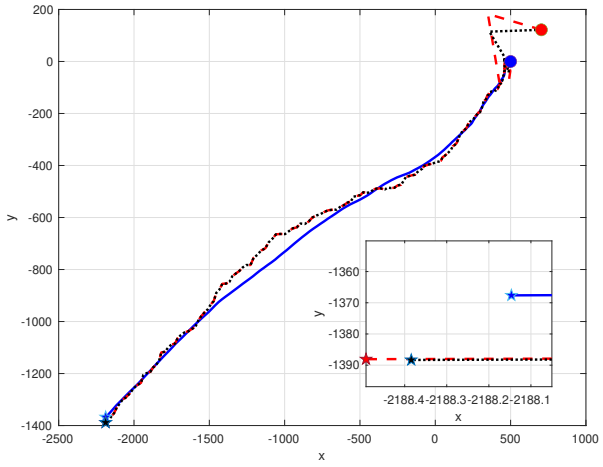

Fig. 2: Estimated trajectories for a representative trial in Scenario 1, which involves a single sensor. The blue, red, and black lines are the target trajectory, the estimated trajectory when $N = 1$, and the estimated trajectory when $N = 10$, respectively. Initial and final states are denoted by circles and stars, respectively, which share the color of the corresponding trajectory.

To assess consistency, 100 independent Monte Carlo trials are performed and the averaged normalized estimation error squared (ANEES) is computed as a performance index. Here the normalized estimation error squared (NEES) is defined as

$$\epsilon(t) = (\hat{x}(t) - x(t))^T \hat{\Lambda}(t)(\hat{x}(t) - x(t))$$

The resulting averaged error trajectories are reported in Fig. 3, and the resulting ANEES behaviors are reported in Fig. 4 for the considered filters.

Fig. 3 and Fig. 4 show the effectiveness of the IBRUF, as

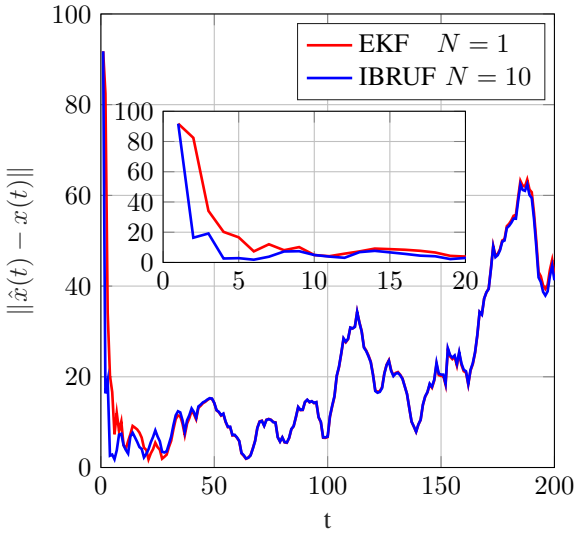

Fig. 3: The averaged error of 100 independent Monte Carlo trials with one sensor's measurements.

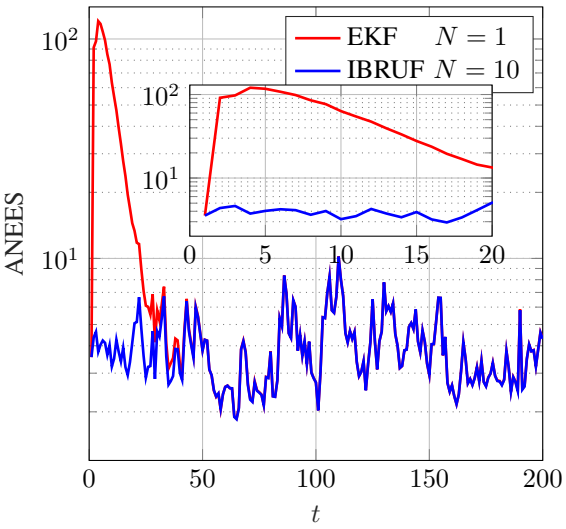

Fig. 4: The ANEES of 100 independent Monte Carlo trials with one sensor's measurements

the case of $N = 10$ outperforms the case of $N = 1$. However, it can be observed that the estimation error increases as t increases even when the ANEES converges. This is caused by the fact that the covariance matrix grows to appropriately reflect the degraded quality of measurements as the target moves far away from the sensor. Without loss of generality, the trajectories for one random trial of the estimation error between the true y coordinate and the estimated y coordinate are plotted in Fig. 5 for $N = 10$. The 3σ bounds are also provided in Fig. 5. This observation indicates that the covariance matrix becomes larger as the target moves away from the sensor. This implies the necessity of multiple sensors for accurate target tracking.

Next, a centralized case where $h(t) = \left[h_1^a(t), h_1^r(t), \ldots, h_6^a(t), h_6^r(t)\right]^T$ is considered. As in

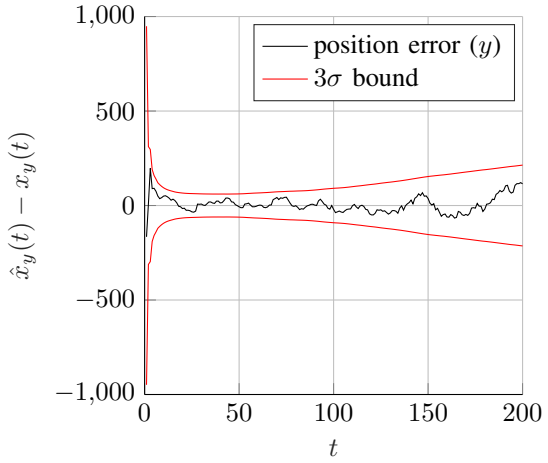

Fig. 5: $N = 10$ with one sensor's measurements. The black curves are the estimation error of the y-coordinate, i.e., $\hat{x}_y(t) - x_y(t)$ for the random trial. The red curves are the 3σ bounds for the random trail.

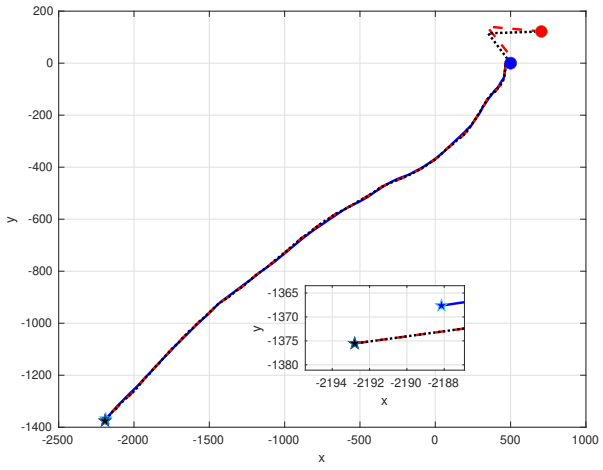

Fig. 6: With six sensors' measurements. The blue, red, and black lines are the target trajectory, the estimated trajectory when $N = 1$, and the estimated trajectory when $N = 10$, correspondingly. The red circle is the initial estimated mean for both $N = 1$ and $N = 10$, and the blue circle is the initial target position. The blue, red, and black stars are the target final position, the estimated value when $N = 1$, and the estimated value when $N = 10$.

the first case, the EKF (equivalent to $N = 1$) and IBRUF with $N = 10$ are considered. The trajectories of the estimates are plotted in Fig. 6. Compared with Fig. 2, the case with six sensors' measurements outperforms the case with only sensor 1's measurements. The resulting averaged error trajectories are reported in Fig. 7, and the corresponding ANEES behaviors are reported in Fig. 8 for the considered filters, respectively.

Scenario 2: Performance of the DIBRUF

For the second scenario, we consider a network consisting of six sensors, where each sensor i has measurements $h_i(t) = \left[h_i^a(t), h_i^r(t)\right]^T$.

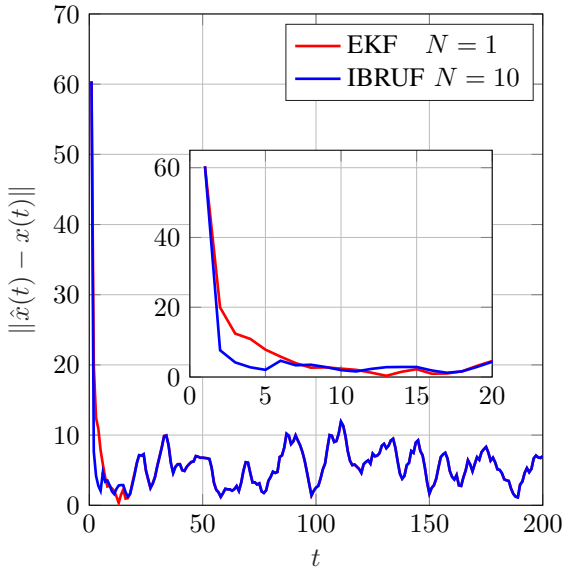

Fig. 7: The averaged error of 100 independent Monte Carlo trials with six sensors' measurements

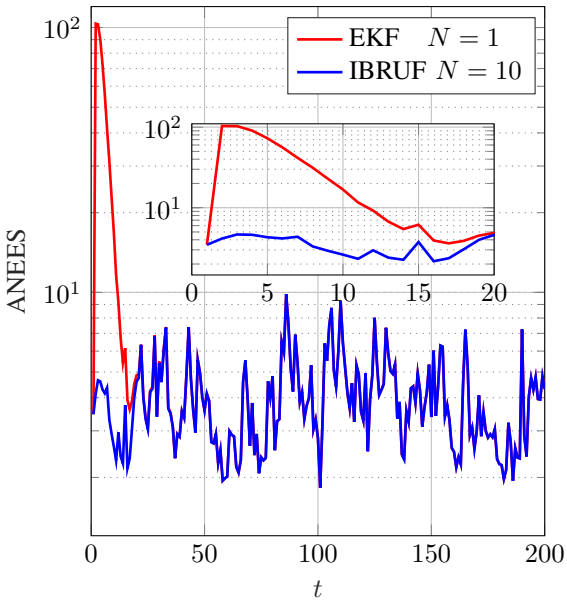

Fig. 8: The ANEES of 100 independent Monte Carlo trials with six sensors' measurements

The network considered is a ring graph as depicted in Fig. 9. Five cases are considered as shown in Table I. For the sake of comparison, 100 independent Monte Carlo trials of Algorithm 2 are performed for each case. The resulting average of the six sensors' averaged error trajectories are reported in Fig. 10, and the corresponding average of the six sensors' ANEES behaviors are reported in Fig. 11 for the considered filters. As can be seen from the insets of Fig. 10 and Fig. 11, all the cases show a stable behavior with or without consensus steps since the system is observable

with one sensor's measurements. However, even minimal communication ($L = 2$) results in significant accuracy improvement over the no-communication ($L = 0$) case.

TABLE I: Filter configurations considered

Filter configuration	N	L
EKF	1	0
DEKF	1	2
DIBRUF$_{2,2}$	2	2
BRUF	10	0
DIBRUF$_{10,2}$	10	2

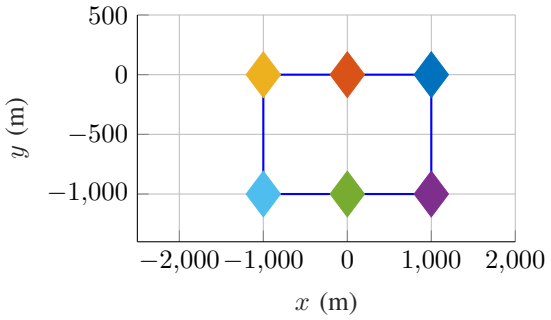

Fig. 9: The ring graph network topology considered in Scenarios 2 and 3.

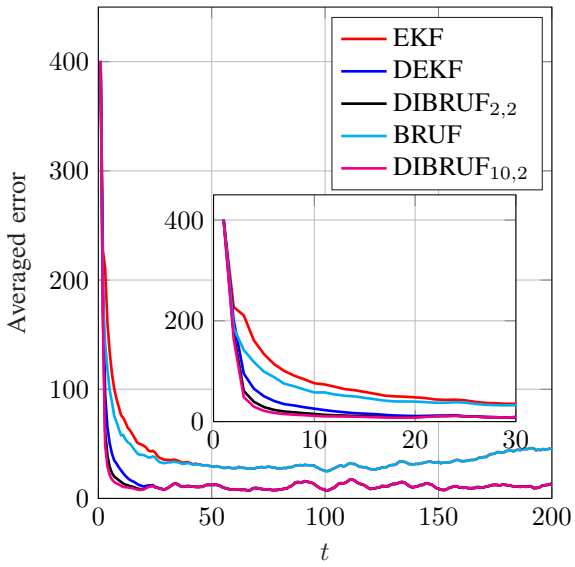

Fig. 10: The averaged error of 100 independent Monte Carlo trials. Here each point on the plot is the mean of six sensors' estimate.

Scenario 3: Necessity of Collective Observability

For the third scenario, we consider a network consisting of six sensors, where each sensor i has one measurement: $h_i(t) = h_i^a(t)$ for $i = 1, 3, 5$, and $h_i(t) = h_i^r(t)$ for $i = 2, 4, 5$. The network considered is a ring graph as depicted in Fig. 9. Five cases are considered as shown in Table I. For the EKF and BRUF cases (in which no communication occurs),

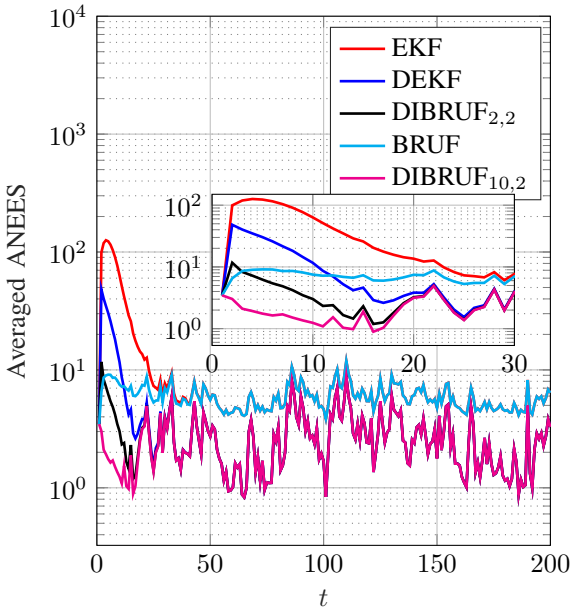

Fig. 11: The ANEES of 100 independent Monte Carlo trials. Here each point on the plot is the mean of six sensors' ANEES.

the system is not observable. For the BRUF case in Table I, Fig. 12 plots the averaged error and the averaged ANEES for $N = 10$. As the system with only one sensor's measurement is not observable, the state can not be recovered, and both the averaged error and the averaged ANEES diverge.

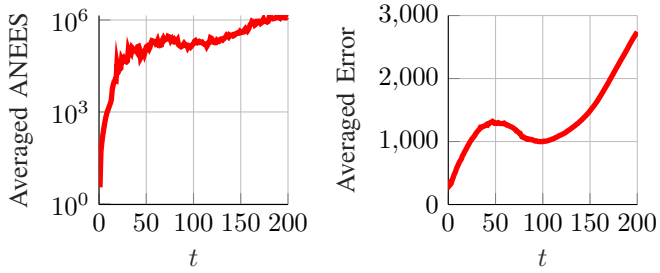

Fig. 12: Averaged performance without communication for a locally unobservable system. Each point corresponds to a value averaged over 100 trials and over six sensors. (a) The averaged error. (b) The averaged ANEES.

For the sake of comparison, 100 independent Monte Carlo trials of Algorithm 2 have been performed for the remaining cases. The resulting average of the six sensors' averaged error trajectories are reported in Fig. 13, and the resulting average of the six sensors' ANEES behaviors are reported in Fig. 14 for the considered filters. As can be seen from the insets of Fig. 13 and Fig. 14, all the cases show a stable behavior with consensus steps since the system is observable with six sensors' measurements. In Fig. 13 and Fig. 14, the green line is the curve of the averaged error and averaged ANEES respectively when the six sensors' measurements

are processed centrally, and Algorithm 1 is run. It can be seen that despite there being no central estimation, the DIBRUF achieves satisfactory performance compared to the ideal centralized case. Furthermore, the $N = 2$ and $N = 10$ DIBRUF configurations produced nearly indistinguishable results, which suggests that DIBRUF's performance advantages are attainable at minimal computational complexity.

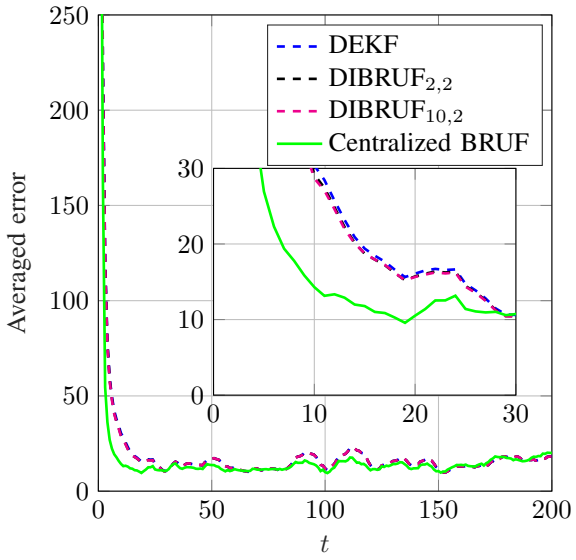

Fig. 13: The averaged error of 100 independent Monte Carlo trials. Here each point on the plot is the mean of six sensors' estimate error.

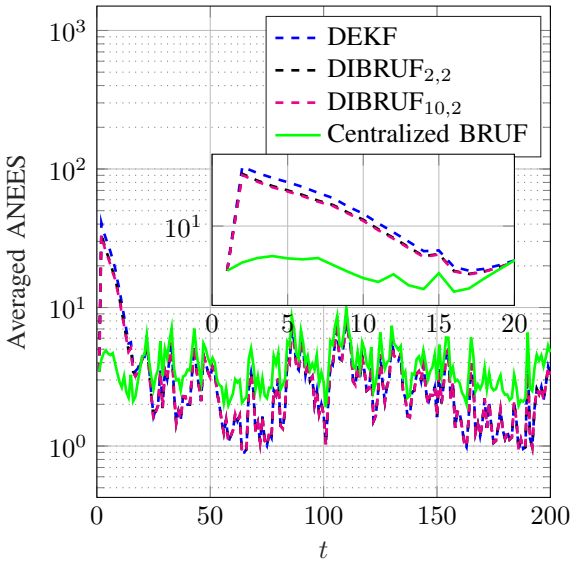

Fig. 14: The ANEES of 100 independent Monte Carlo trials. Here each point on the plot is the mean of The ANEES of 100 independent Monte Carlo trials. Here each point on the plot is the mean of six sensors' ANEES.

V. Conclusion

This paper presents the distributed update scheme for nonlinear measurements called the distributed information Bayesian recursive update filter (DIBRUF). Compared to the distributed extended Kalman filter, the DIBRUF reduces the linearization error of the extended Kalman filter by dividing the measurement update into N steps. Compared to the Bayesian recursive update filter and information Bayesian recursive update filter, which both require the system to be observable with a central measurement, the DIBRUF requires the network to be only collectively observable as the sensors can share information among the network. This combination of recursive nonlinear filtering and distributed estimation finds applications in sensor networks, multi-robot systems, distributed control, and decentralized decision-making, among others. Ongoing research in this area focuses on developing more efficient and resilient algorithms to handle large-scale distributed systems with diverse communication and computation constraints.

References

[1] Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.

[2] Renato Zanetti. Recursive update filtering for nonlinear estimation. *IEEE Transactions on Automatic Control*, 57(6):1481–1490, 2012.

[3] Kristen Michaelson, Andrey A. Popov, and Renato Zanetti. Bayesian recursive update for ensemble Kalman filters. *arXiv*, 2023.

[4] Kristen Michaelson, Andrey A. Popov, and Renato Zanetti. Ensemble Kalman filter with Bayesian recursive update. In *2023 26th International Conference on Information Fusion (FUSION)*, pages 1–6, 2023.

[5] Reza Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, pages 8179–8184, 2005.

[6] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5492–5498, 2007.

[7] U. A. Khan and Ali Jadbabaie. On the stability and optimality of distributed Kalman filters with finite-time data fusion. In *Proceedings of the 2011 American Control Conference*, pages 3405–3410, 2011.

[8] Reza Olfati-Saber. Kalman-Consensus Filter : Optimality, stability, and performance. In *Proceedings of the 48h IEEE Conference on Decision and Control, and the 28th Chinese Control Conference*, pages 7036–7042, 2009.

[9] Usman A Khan, Soummya Kar, Ali Jadbabaie, and José M.F. Moura. On connectivity, observability, and stability in distributed estimation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6639–6644, 2010.

[10] Giorgio Battistelli and Luigi Chisci. Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50:707–718, 3 2014.

[11] Giorgio Battistelli and Luigi Chisci. Stability of consensus extended Kalman filter for distributed state estimation. *Automatica*, 68:169–178, 6 2016.

[12] Giorgio Battistelli, Luigi Chisci, Daniela Selvi, and Pietro Tesi. Distributed state estimation under denial of service. In *2019 IEEE 58th Conference on Decision and Control*, pages 7818–7823, 2019.

[13] Giorgio Battistelli, Luigi Chisci, and Daniela Selvi. A distributed Kalman filter with event-triggered communication and guaranteed stability. *Automatica*, 93:75–82, 2018.

[14] Giorgio Battistelli, Luigi Chisci, Daniela Selvi, and Pietro Tesi. Stability of consensus-based distributed estimation under denial of service. *IEEE Transactions on Automatic Control*, pages 1–8, 2023.

[15] Hao Chen, Jianan Wang, Chunyan Wang, Jiayuan Shan, and Ming Xin. Distributed diffusion unscented Kalman filtering based on covariance intersection with intermittent measurements. *Automatica*, 132:109769, 2021.

[16] Shauying R. Kou, David L. Elliott, and Tzyh Jong Tarn. Observability of nonlinear systems. *Information and Control*, 22:89–99, 1973.