Distributed Bayesian Filter using Measurement Dissemination for Multiple UGVs with Dynamically Changing Interaction Topologies

Chang Liu
Department of Mechanical Engineering
University of California, Berkeley
Berkeley, CA 94720
Email: changliu@berkeley.edu

Shengbo Eben Li*
Department of Automotive Engineering
Tsinghua University
Beijing, China 100084
Email: lisb04@gmail.com

J. Karl Hedrick
Department of Mechanical Engineering
University of California, Berkeley
Berkeley, CA 94720
Email: khedrick@me.berkeley.edu

This paper presents a novel distributed Bayesian filtering algorithm using measurement dissemination for multiple unmanned ground vehicles (UGVs) with dynamically changing interaction topologies. Different from statistics dissemination-based approaches that transmit posterior distributions or likelihood functions, this algorithm relies on the measurement dissemination scheme, which significantly reduces the transmission burden between each pair of UGVs. Each UGV only sends a communication buffer and a track list to neighboring agents. A communication buffer contains its own measurements and the received ones from other UGVs. And a track list is used to trim all measurements in the buffer to reduce communication overhead. Under the condition that the interaction topologies are frequently jointly strongly connected, each UGV's measurements can be disseminated over the network within finite time. The proposed DBF algorithm is proved to achieve consistent state estimation. The effectiveness of this algorithm is validated by comparing with consensus-based distributed filters and the centralized filter in multi-target tracking simulations.

1 INTRODUCTION

Distributed estimation using a group of networked UGVs has been utilized to collectively infer environment status, such as intruder detection [1] and object tracking [2]. Several techniques have been developed for distributed estimation, including distributed linear Kalman filter (DKF) [3], distributed extended Kalman filter [4], and distributed particle filter [5]. The most generic filtering scheme is the distributed Bayesian filter (DBF), which can be applied for nonlinear systems with arbitrary noise distributions [6, 7]. In fact, under the assumption of linearity and Gaussian noise, a DBF can be reduced to a DKF [8]. This paper focuses on the DBF for networked UGVs under dynamically changing interaction topologies.

The interaction topology plays a central role on the design of DBF, of which two types are widely investigated in literature: fusion center (FC) and neighborhood (NB). In the former, local statistics estimated by each agent is transmitted to a single FC, where a global posterior distribution is calculated at each filtering cycle [9,10]. In the latter, each agent individually executes distributed estimation and the agreement of local estimates is achieved by certain consensus strategies [11–13].In general, the NB-based distributed filters are more suitable in practice since they do not require a fusion center with powerful computation capability and are more robust to changes in network topology and link failures. So far, the NB-based approaches have two mainstream schemes according to the transmitted data among agents, i.e., *statistics dissemination* (SD) and *measurement dissemination* (MD). In the SD scheme, each agent exchanges statistics such as posterior distributions and likelihood functions within neighboring nodes [14]. In the MD scheme, instead of exchanging statistics, each agent sends measurements to neighboring nodes.

Statistics dissemination scheme has gained increasing interest and been widely investigated during last decade. Madhavan et al. (2004) presented a distributed extended Kalman filter for nonlinear systems [4]. This filter was used to generate local terrain maps by using pose estimates to combine elevation gradient and vision-based depth with environmental features. Olfati-Saber (2005) proposed a dis-

*Address all correspondence to this author.

tributed linear Kalman filter (DKF) for estimating states of linear systems with Gaussian process and measurement noise [3]. Each DKF used additional low-pass and band-pass consensus filters to compute the average of weighted measurements and inverse-covariance matrices. Gu (2007) proposed a distributed particle filter for Markovian target tracking over an undirected sensor network [5]. Gaussian mixture models (GMM) were adopted to approximate the posterior distribution from weighted particles and the parameters of GMM were exchanged via average consensus filter. Hlinka et al. (2012) proposed a distributed method for computing an approximation of the joint (all-sensors) likelihood function by means of weighted-linear-average consensus algorithm when local likelihood functions belong to the exponential family of distributions [15]. Saptarshi et al. (2014) presented a Bayesian consensus filter that uses logarithmic opinion pool for fusing posterior distributions of the tracked target [6]. Other examples can be found in [7] and [16].

Despite the popularity of statistics dissemination, exchanging statistics can consume high communication resources. One remedy is to approximate statistics with parametric models, e.g., Gaussian Mixture Model [17], which can reduce communication burden to a certain extent. However, such manipulation increases the computation burden of each agent and sacrifices filtering accuracy due to approximation. The measurement dissemination scheme is an alternative solution to address the issue of exchanging statistics. An early work on measurement dissemination was done by Coates et al. (2004), who used adaptive encoding of observations to minimize communication overhead [18]. Ribeiro et al. (2006) exchanged quantized observations along with error-variance limits considering more pragmatic signal models [19]. A recent work was conducted by Djuric et al. (2011), who proposed to broadcast raw measurements to other agents, and therefore each agent has a complete set of observations of other agents for executing particle filtering [20]. A shortcoming of aforementioned works is that their communication topologies are assumed to be a fixed and complete graph that every pair of distinct agents is constantly connected by a unique edge. In many real applications, the interaction topology may change dynamically due to unreliable links, external disturbances and/or range limits [21]. In such cases, dynamically changing topologies can cause random packet loss, variable transmission delay, and out-of-sequence measurement issues (OOSM) [22], thus decreasing the performance of distributed estimation. Leung et al. (2010) explored a decentralized Bayesian filter for dynamic robot networks [23]. The algorithm was shown to achieve centralized-equivalent filtering performance in simulations. However, it required the communication of both measurements and statistics, which could incur large communication overhead.

The main contribution of the paper is that we present a measurement dissemination-based distributed Bayesian filtering (DBF) method for a group of networked UGVs with dynamically changing interaction topologies. In our previous work [24], we have proposed a Latest-In-and-Full-Out (LIFO) protocol for data exchange and developed a LIFO-based DBF. However, it only applies to static target with simple binary sensor model. In this work, we introduce the concept of the track list and extend our methods to time-varying topologies and more general sensor models. The measurement dissemination scheme uses the so-called Full-In-and-Full-Out (FIFO) protocol, under which each UGV is only allowed to broadcast observations to its neighbors by using single-hopping. An individual Bayesian filter is implemented locally by each UGV after exchanging observations using FIFO. Under the condition that the topologies is frequently jointly strongly connected, FIFO can disseminate measurements over the network within a finite time. The main benefit of using FIFO is on the reduction of communication burden while avoiding the OOSM issue and ensuring that no information loss occurs.

The rest of this paper is organized as follows: Section 2 formulates the target tracking problem using multiple UGVs; Section 3 proposes the FIFO protocol for measurement communication in dynamically changing interaction topologies; Section 4 introduces FIFO-based DBF algorithm and track list; simulation results are presented in Section 6 and Section 7 concludes the paper.

2 Problem Formulation

Consider a network of N UGVs in a bounded two-dimensional space S. The interaction topology can be dynamically changing due to limited communication range, varying team formation or link failure. Each UGV is equipped with a sensor for target detection. Due to the limit of communication range, each UGV can only exchange sensor measurements with its local neighbors. Every UGV locally runs a Bayesian filter to estimate the target position in S utilizing its own measurements and the received measurements from other UGVs. Since this work is focused on the distributed Bayesian filter for target localization, we assume that UGVs' states are accurately known.

2.1 Target and Sensor Model

The target motion uses a stochastic discrete-time model:

$$x^g_{k+1} = f(x^g_k, v_k), \tag{1}$$

where the superscript g represents the target and $x^g_k \in S$ is the target position at time k; v_k is the white process noise.

The sensor measurement is described by a stochastic model:

$$z^i_k = h_i(x^g_k, x^i_k, w^i_k), \tag{2}$$

where the superscript $i \in \{1, \dots, N\}$ represents the index of the UGV; $x^i_k \in S$ is the sensor position and w^i_k is the white measurement noise. The measurement function h_i depends on the type of the sensor.

The conditional probability, $P(z^i_k|x^g_k;x^i_k)$, of obtaining a certain measurement z^i_k conditioning on the target and sensor states is critical to designing the Bayesian filter [25]. It also depends on the distribution of the measurement noise. For example, if w^i_k is an additive, zero-mean Gaussian white noise with covariance Γ^i_k, then, according to Eq. (2),

$P(z^i_k|x^g_k;x^i_k)$ can be described as

$$P(z^i_k|x^g_k;x^i_k) = \mathcal{N}(h_i(x^g_k,x^i_k),\Gamma^i_k).$$

For non-Gaussian noise distributions, such as the Poisson noise or Cauchy noise [26], $P(z^i_k|x^g_k;x^i_k)$[1] can also be similarly defined. It should be noted that, the approach presented in this work does not rely on the specific distribution of the noise.

The h_i for several typical sensors are defined as follows [27]:

Range-only sensors: h_i only depends on the relative Euclidean distance between the sensor and the target:

$$h_i(x^g_k,x^i_k) = \|x^g_k - x^i_k\|_2,$$

where $\|\cdot\|_2$ is the Euclidean distance in S.

Bearing-only sensors: h_i only depends on the relative bearing between the sensor and the target:

$$h_i(x^g_k,x^i_k) = \measuredangle(x^g_k - x^i_k),$$

where $\measuredangle$ denotes the angle from the sensor to the target.

Range-bearing sensors: h_i includes both the relative distance and bearing:

$$h_i(x^g_k,x^i_k) = x^g_k - x^i_k.$$

Remark 1. *Given the knowledge of current target and UGV positions, the current measurement by each UGV can be considered conditionally independent from its own past measurements and those by other UGVs [28].*

2.2 Graphical Model of Interaction Topology

We consider a simple[2] graph $G = (V,E)$ to represent the interaction topology of N networked UGVs, where the vertex set $V = \{1,\dots,N\}$ represents the index set of UGVs and $E = V \times V$ denotes the edge set. For the purpose of clarity and generalizability, we use directed graphs to describe our approach in this work. However, the approach can conveniently apply to undirected graphs, which can be treated as bidirectional directed graphs.

The *adjacency matrix* $A = [a_{ij}]$ of the graph G describes the interaction topology:

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E \end{cases},$$

where a_{ij} is the entry on the i^{th} row and j^{th} column of the adjacency matrix. The notation $a_{ij} = 1$ indicates that the i^{th} UGV can directly communicate to the j^{th} UGV and $a_{ij} = 0$ indicates no direct communication from i to j. A directed graph is *strongly connected* if there is a directed path connecting any two arbitrary vertices in V[3].

[1] For the purpose of simplicity, we will not explicitly write the parameter x^i_k in $P(z^i_k|x^g_k;x^i_k)$ for the rest of the paper.

[2] A (directed/undirected) graph $G = (V,E)$ is *simple* if it has no self-loops (i.e., $(i,j) \in E$ only if $i \neq j$) or multiple edges with the same source and target nodes (i.e., E only contains distinct elements).

[3] An undirected graph with this property is called a *connected* graph.

Let $\bar{G}$ denote the set of all possible simple directed graphs[4] defined over the network of UGVs. The adjacency matrix associated with a graph $G_l \in \bar{G}$ is denoted as $A_l = [a^l_{ij}]$. Define the *union* of a collection of graphs $\left\{G_{i_1}, G_{i_2}, \dots, G_{i_l}\right\} \subset \bar{G}$ as the graph with the vertices in V and the edge set given by the union of edge sets of G_{i_j}, $j = 1\dots,l$. Such collection is *jointly strongly connected* if the union of its members forms a strongly connected graph[5]. We define the *direct neighbors* of i^{th} UGV under topology G_l as the set $\mathcal{N}_i(G_l) = \left\{j|a^l_{ij} = 1, j \in V\right\}$. All UGVs in $\mathcal{N}_i(G_l)$ can directly receive information from the i^{th} UGV via the single hopping.

3 Full-In-and-Full-Out (FIFO) Protocol

This study proposes a Full-In-and-Full-Out (FIFO) protocol for measurement exchange in time-varying topologies. Let $Y^i_{\mathcal{K}} = \left\{\left[x^i_k, z^i_k\right] | k \in \mathcal{K}\right\}$ be the set of state-measurement pairs of robot i, where $\mathcal{K}$ is an index set of time steps. Each UGV contains a communication buffer (CB) that stores state-measurement pairs of all UGVs:

$$\mathcal{B}^i_k = \left[Y^1_{\mathcal{K}^{i,1}_k}, \dots, Y^N_{\mathcal{K}^{i,N}_k}\right],$$

where $\mathcal{B}^i_k$ is the CB of i^{th} UGV at time k and $\mathcal{K}^{i,j}_k (j \in V)$ is the time index set. $Y^j_{\mathcal{K}^{i,j}_k}$ represents the set of j^{th} UGV's measurements at time steps in $\mathcal{K}^{i,j}_k$ that are stored in i^{th} UGV's CB at time k. The **FIFO protocol** is stated in Algorithm 1. Note that each section in the algorithm contains the CB and TL parts. For the purpose of clarity, we ignore the TL parts at this stage and will describe them in Section 4.2.

Fig. 1 illustrates the FIFO cycles of a network of 3 UGVs with switching topologies. Several facts can be noticed in Fig. 1: (1) the topologies are jointly strongly connected in the time interval $[1,6]$; (2) CBs of all UGVs are filled within finite steps; Extending these facts to a network of N UGVs, we have the following theorem to describe the property of FIFO:

Theorem 1. *Consider a network of N UGVs with switching interaction topologies. If the following two conditions hold:*

1. *there exists an infinite sequence of time intervals $[k_m, k_{m+1})$, $m = 1,2,\dots,$ starting at $k_1 = 0$ and are contiguous, nonempty and uniformly bounded;*
2. *the union of graphs across each such interval is jointly strongly connected,*

then any pair of UGVs can exchange measurements under FIFO. And the communication delay between each pair of UGVs is no greater than $(N-1)T_u$, where $T_u = \sup_{m=1,2,\dots} (k_{m+1} - k_m)T$ is the upper bound of interval lengths.

Proof. Without loss of generality, we consider the transmission of B^i_1 from the i^{th} UGV to an arbitrary j^{th} UGV

[4] For undirected graphs, we consider the set of simple undirected graphs.

[5] For undirected graphs, such collection is *jointly connected* [11].

Algorithm 1 FIFO Protocol

(1) Initialization.

CB: The CB of i^{th} UGV is initialized as an empty set at $k = 0$:

$$\mathcal{B}_0^i = \left[Y^1_{\mathcal{K}_0^{i1}}, \ldots, Y^N_{\mathcal{K}_0^{iN}}\right], \text{ where } Y^j_{\mathcal{K}_0^{ij}} = \{[\varnothing, \varnothing]\}.$$

TL: The TL of i^{th} UGV is initialized at $k = 0$:

$$P_0^i = \mathbf{0}, \text{ i.e. } p_0^{jl} = 0, \forall j, l \in \{1 \ldots, N\}.$$

(2) At time $k\,(k \geq 1)$ for i^{th} UGV:

(2.1) Receiving Step.

CB: The i^{th} UGV receives all CBs of its direct neighbors $\mathcal{N}_i(G_{k-1})$, each corresponding to the $(k-1)^{\text{th}}$ step CB of a UGV in $\mathcal{N}_i(G_{k-1})$. The received CB from the l^{th} UGV is

$$\mathcal{B}_{k-1}^l = \left[Y^1_{\mathcal{K}_{k-1}^{l1}}, \ldots, Y^N_{\mathcal{K}_{k-1}^{lN}}\right], \, l \in \mathcal{N}_i(G_{k-1})$$

TL: The i^{th} UGV receives all TLs of its direct neighborhood $\mathcal{N}_i(G_{k-1})$. The received TL from the l^{th} UGV is Q_{k-1}^l.

(2.2) Observation Step.

CB: The i^{th} UGV updates $Y^i_{\mathcal{K}_k^{i,i}}$ by its own state-measurement pair at current step:

$$Y^i_{\mathcal{K}_k^{ii}} = Y^i_{\mathcal{K}_{k-1}^{ii}} \cup \left\{\left[x_k^i, z_k^i\right]\right\}.$$

(2.3) Updating Step.

CB: The i^{th} UGV updates other elements of its own CB, $Y^j_{\mathcal{K}_k^{ij}}\,(j \neq i)$, by merging with all received CBs:

$$Y^j_{\mathcal{K}_k^{ij}} = Y^j_{\mathcal{K}_{k-1}^{ij}} \cup Y^j_{\mathcal{K}_{k-1}^{lj}}, \, \forall j \neq i, \, \forall l \in \mathcal{N}_i(G_{k-1}).$$

TL: The i^{th} UGV updates its own TL, Q_k^i, using the received TLs (see Algorithm 3). Trim the CB based on the updated track lists (see Algorithm 4).

(2.4) Sending Step:

CB: The i^{th} UGV broadcasts its updated CB, $\mathcal{B}_k^i = \left[Y^1_{\mathcal{K}_k^{i1}}, \ldots, Y^N_{\mathcal{K}_k^{iN}}\right]$, to all of its neighbors defined in $\mathcal{N}_i(G_k)$.

TL: The i^{th} UGV broadcasts its updated track list to its neighbors $\mathcal{N}_i(G_k)$.

(3) $k \leftarrow k+1$ until stop

$(j \in V \setminus \{i\})$. Since each UGV will receive direct neighbors' CBs and send the merged one to its neighbors at the next time step, the i^{th} UGV can transmit B_1^i to j if and only if there is a path from vertex i to j. As the union of graphs across the time interval $[k_1, k_2)$ is jointly connected, i^{th} UGV can directly send B_1^i to at least one another UGV at a time instance, i.e., $\exists l_1 \in V \setminus \{i\}, \exists t_1 \in [k_1, k_2)$ s.t. $l_1 \in \mathcal{N}_i(G_{t_1})$. If $l_1 = j$, then B_1^i has been sent to j. If $l_1 \neq j$, B_1^i has been merged into $B_{t_1}^{l_1}$ and will be sent out in the next time step.

By using the similar derivation for time intervals $[k_m, k_{m+1})$, $, m = 2, 3, \ldots,$ it can be shown that all UGVs can receive the state-measurement pairs in B_1^i no later by k_N. Therefore, the transmission delay between an arbitrary pair of UGVs is no greater than $(N-1)T_u$.

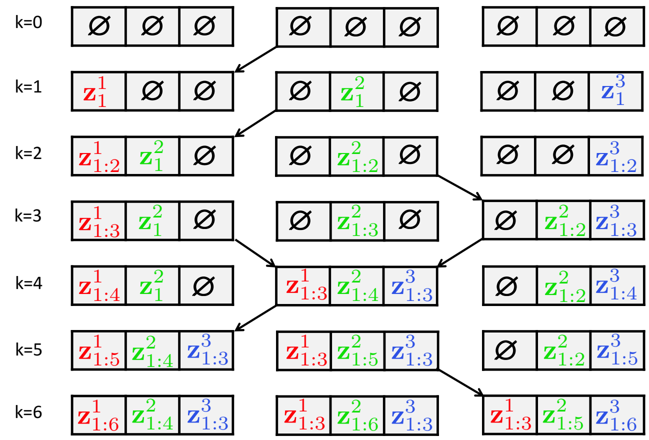


Fig. 1: Example of FIFO with three UGVs using switching line interaction topologies. The double-headed arrow represents a communication link between two UGVs.

Similar to the definition in [11], we define an interaction topology that satisfies the two conditions in Theorem 1 as a *frequently jointly strongly connected* network.

Corollary 1. *For a frequently jointly strongly connected network, each UGV receive the CBs of all other UGVs under FIFO within finite time.*

Proof. According to Theorem 1, each UGV is guaranteed to receive B_t^j, $\forall t \geq 0$, $j \in V$ when $k \geq t + (N-1)T_u$.

Remark 2. *The frequency that each UGV receives other UGVs' CBs depends on the property of the network. Corollary 1 gives an upper bound on the transmission time for all frequently jointly strongly connected networks under FIFO.*

4 Distributed Bayesian Filter via FIFO Protocol

We first introduce the generic distributed Bayesian filter (DBF). Let $X_k \in S$ be the random variable representing the position of the target at time k. Define $\mathcal{Z}_k^i$ as the set of the measurements of time k from all UGVs that are in the i^{th} UGV's CB, i.e., $\mathcal{Z}_k^i = \left\{z_k^j \middle| \left[x_k^j, z_k^j\right] \in \mathcal{B}_k^i, \, \forall j \in V\right\}$and let $\mathcal{Z}_{1:k}^i = \bigcup_{t=1}^{k} \mathcal{Z}_t^i$. To avoid confusion, we define $\mathbf{z}_{1:k}^i = \left[z_1^i, \ldots, z_k^i\right]$ as the set of the i^{th} UGV's measurements of times 1 through k. The probability density function (PDF) of X_k, called *individual PDF*, of the i^{th} UGV is represented by $P_{pdf}^i(X_k|\mathcal{Z}_{1:k}^i)$ and is the estimation of the target position given all the measurements that the i^{th} UGV has received. The initial individual PDF, $P_{pdf}^i(X_0)$, is constructed given prior information including past experience and environment knowledge. It is necessary to initialize $P_{pdf}^i(X_0)$ such that the probability density of the true target position is nonzero, i.e., $P_{pdf}^i(X_0 = x_0^g) > 0$.

Algorithm 2 FIFO-DBF Algorithm

For i^{th} UGV at k^{th} step ($\forall i \in V$):

(1) Initialize a *temporary PDF* by assigning the stored individual PDF to it:

$$P^i_{tmp}(X_t) = P^i_{sto}(X_t),$$

where

$$P^i_{sto}(X_t) = P^i_{pdf}(X_t|z^1_{1:t}, \ldots, z^N_{1:t}).$$

(2) For $\xi = t+1$ to k, iteratively repeat two steps of Bayesian filtering:

(2.1) Prediction

$$P^{pre}_{tmp}(X_\xi) = \int_S P(X_\xi|X_{\xi-1}) P^i_{tmp}(X_{\xi-1}) dX_{\xi-1}.$$

(2.2) Updating

$$P^i_{tmp}(X_\xi) = K_\xi P^{pre}_{tmp}(X_\xi) P(\mathcal{Z}^i_\xi|X_\xi),$$

$$K_\xi = \left[\int_S P^{pre}_{tmp}(X_\xi) P(\mathcal{Z}^i_\xi|X_\xi) dX_\xi\right]^{-1}.$$

(2.3) When $\xi = t+1$, if $z^j_{t+1} \neq \emptyset$ for $\forall j \in V$, update the stored PDF:

$$P^i_{sto}(X_{t+1}) = P^i_{tmp}(X_{t+1}).$$

(3) The individual PDF of i^{th} UGV at time k is $P^i_{pdf}(X_k|\mathbf{z}^i_{1:k}) = P^i_{tmp}(X_k)$.

Under the framework of DBF, the individual PDF is recursively estimated by two steps: the prediction step and the updating step.

Prediction. At time k, the prior individual PDF $P^i_{pdf}(X_{k-1}|\mathcal{Z}^i_{1:k-1})$ is first predicted forward by using the Chapman-Kolmogorov equation:

$$P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1}) = \int\limits_{X_{k-1}\in S} P(X_k|X_{k-1}) P^i_{pdf}(X_{k-1}|\mathcal{Z}^i_{1:k-1}) dX_{k-1}, \tag{3}$$

where $P(X_k|X_{k-1})$ represents the state transition probability of the target, based on the Markovian motion model (Eq. (1)).

Updating. The i^{th} individual PDF is then updated by the Bayes' rule using the set of newly received measurements at time k, i.e., $\mathcal{Z}^i_k$:

$$\begin{aligned} P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k}) &= K_i P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1}) P(\mathcal{Z}^i_k|X_k) \\ &= K_i P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1}) \prod_{j\in\Omega^i_k} P(z^j_k|X_k) \end{aligned} \tag{4}$$

where $P(z^j_k|X_k)$ is the sensor model and K_i is a normalization factor, given by:

$$K_i = \left[\int\limits_{X_k\in S} P^i_{pdf}(X_k|\mathbf{z}^i_{1:k-1}) P(\mathcal{Z}^i_k|X_k) dX_k\right]^{-1}.$$

The Ω^i_k denotes the index set of UGVs whose state-measurement pair of time k is stored in the i^{th} UGV's CB, i.e. $\Omega^i_k = \left\{ j \in V | \left[x^j_k, z^j_k\right] \in \mathcal{B}^i_k, \forall j \in V \right\}$. The factorization of $P(\mathcal{Z}^i_k|X_k)$ comes from the conditional independence of measurements between each UGV given the target position and the corresponding UGV's position.

4.1 FIFO-DBF

The generic DBF is not directly applicable to time-varying interaction topologies. This is because changing topologies can cause intermittent and out-of-sequence reception of measurements from different UGVs, which can give rise to the OOSM problem. One possible solution is to ignore all measurements that are out of the temporal order. This is, however, undesirable since this will cause information loss. Another possible remedy is to incorporate all measurements from the initial individual PDF at every time step. This, however, causes excessive computational burden. To avoid the OOSM problem and remove unnecessary computational complexity, we add a new PDF, call the *stored PDF*. The stored PDF, $P^i_{sto}(X_t)$, is updated from the i^{th} UGV's initial PDF by fusing the state-measurement pairs of *all* UGVs up to a certain time $t \leq k$. The choice of t is described in Section 4.2. The individual PDF, $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k})$, is then computed by fusing the measurements from time $t+1$ to k in the CB into $P^i_{sto}(X_t)$, running the Bayesian filter (Eq. (3) and (4)). Note that initially, $P^i_{sto}(X_0) = P^i_{pdf}(X_0)$.

The **FIFO-DBF algorithm** is stated in Algorithm 2. Each UGV runs FIFO-DBF after its CB is updated in the Updating Step in Algorithm 1. At the beginning, we assign the stored PDF to a temporary PDF, which will then be updated by sequentially fusing measurements in the CB to obtain the individual PDF. It should be noted that, when the UGV's CB contains all UGVs' state-measurement pairs of time $t+1$, the temporary PDF of $t+1$ is assigned as the stored PDF. Fig. 2 illustrates the FIFO-DBF procedure for the 1$^{\text{st}}$ UGV as an example. It can be noticed that, the purpose of using the stored PDF is to avoid running the Bayesian filtering from the initial PDF at every time step. Since the stored PDF has incorporated all UGVs' measurements up to some time step t, the information loss problem no longer exists. We point out that the time t of each UGV's stored PDF can be different from others. The stored PDF is saved locally by each UGV and not transmitted to others.

Algorithm 3 Updating TLs

Consider updating the i^{th} UGV's TL using the received j^{th} UGV's TL.

(1) Update the i^{th} row of TL using the i^{th} UGV's CB (Eq. (5)).

(2) Update the j^{th} row of TL using the received j^{th} UGV's CB $\mathcal{B}^j_{k-1}$.

(3) Update other rows of TL: $\forall l \in \mathcal{N}_i(G_{k-1}) \setminus \{j\}, \forall r \in \{1 \ldots, N\}$,

if $k^{ir} > k^{lr}$, keep current $\mathbf{q}^{ir}_{k^{ir}}$;

if $k^{ir} = k^{lr}$, $\mathbf{q}^{ir}_{k^{ir}} = \mathbf{q}^{ir}_{k^{ir}} \vee {}^6\mathbf{q}^{lr}_{k^{lr}}$;

if $k^{ir} < k^{lr}$, $\mathbf{q}^{ir}_{k^{ir}} = \mathbf{q}^{lr}_{k^{lr}}$ and $k^{ir} = k^{lr}$.

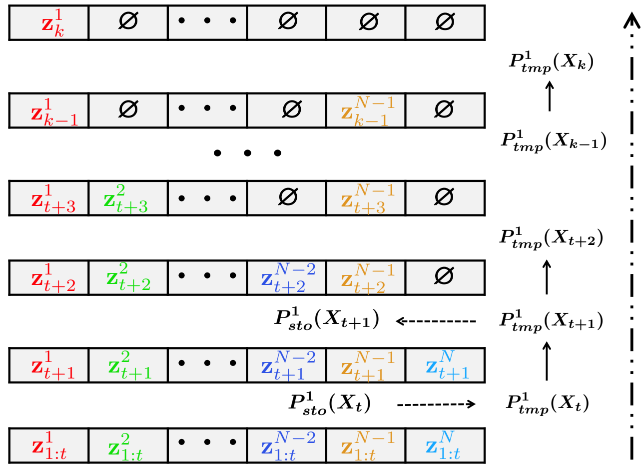


Fig. 2: Example of FIFO-DBF for 1st UGV at time k. Networked UGVs take a line topology. The stored individual PDF is represented by $P^1_{pdf}(k-N)$. The UGV first calculates $P^1_{tmp}(k-N+1)$, defined in Algorithm 2, and then stores it as $P^1_{pdf}(k-N+1)$. Repeating DBF until obtaining $P^1_{pdf}(k)$. In this example, $\Omega^1_\xi = \{1,2,\ldots,N+1-\xi\}$, $\xi = 1,\ldots,N$.

4.2 Track Lists for Trimming Communication Buffers

Algorithm 4 Trimming CBs using TLs

For the i^{th} UGV: find the smallest time in the track list: $k^i_m = \min\left\{k^1,\ldots,k^N\right\}$. If all entries associated with time k^i_m in Q^i_k are 1's, then

1. set all these items in the track list to be 0, let $k_m \leftarrow k_m+1$.
2. update the track list with the measurements in the current CB, i.e., $q^{jl}_{k^{ij}} = 1$ if and only if $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right] \in \mathcal{B}^i_k$.
3. remove all corresponding measurements in the i^{th} UGV's CB.

The size of CBs can keep increasing as measurements cumulate over time. The use of the stored PDF has made it feasible to trim excessive measurements from the CBs. To avoid information loss, a state-measurement pair can be trimmed from a UGV's CB only when *all* UGVs have received it. To keep track of each UGV's reception of other UGVs' measurements, every UGV maintains a *track list* (TL), $Q^i_k = \left[\mathbf{q}^{i1}_{k^{i1}},\ldots,\mathbf{q}^{iN}_{k^{iN}}\right]^T$ $(\forall i \in V)$, where $\mathbf{q}^{ij}_{k^{ij}} = \left[q^{jl}_{k^{ij}}, l \in V\right]$ is a binary vector with size N. Therefore, a TL can be represented by a binary matrix with the size $N\times N$. For the i^{th} UGV, the Q^i_k represents its knowledge of the earliest measurements, in terms of the measurement time, by all UGVs. Each element $q^{jl}_{k^{ij}}$ equals 1 if the i^{th} robot knows that the state-measurement pair of the l^{th} UGV of time k^{ij}, $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right]$, has been received by the j^{th} robot, and equals 0 if the i^{th} robot cannot determine whether $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right]$ has been received by the j^{th} robot, i.e.,

$$q^{jl}_{k^{ij}} = \begin{cases} 1 & \text{if} \quad \exists t \in \left[k^{ij}, k\right] \text{ s.t. } \left[x^l_{k^{ij}}, z^l_{k^{ij}}\right] \in \mathcal{B}^j_t, \\ 0 & \text{if} \quad \nexists t \in \left[k^{ij}, k\right] \text{ s.t. } \left[x^l_{k^{ij}}, z^l_{k^{ij}}\right] \in \mathcal{B}^j_t. \end{cases} \tag{5}$$

It can happen that $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right]$ has been received by the j^{th} UGV but the i^{th} UGV does not know this and thus $q^{jl}_{k^{ij}} = 0$.

The exchange and updating of TLs are described in the TL part in Algorithms 1 and 3. The Algorithm 4 describes the approach to trim CBs using TLs. Since a TL only keeps track of the earliest measurements in CBs, the CBs are only trimmed by one time step each time the trim happens. (**TODO:** add a plot showing the change of TL.) (**TODO:** think how to illustrate the trim process?)

The use of TLs can avoid the excessive size of CBs and guarantee that trimming the CBs will not lose any information; the trimmed measurements have been encoded into the stored PDF. The following theorem formalizes this property.

Theorem 2. *For a frequently jointly strongly connected network using FIFO-DBF, each UGV's estimation result using the trimmed CB by the TL is the same as that using the non-trimmed CB.*

Proof. Let $k_m = \min_j k^{ij}$. Trimming Q^i_k happens when all entries are 1. This indicates that each UGV has received the state-measurement pairs of time k_m from all UGVs. A UGV has either saved the pairs in its CB or already fused them to update its individual PDF. In both cases, such pairs are no longer needed to be transmitted since it will not add any unused information to the multi-agent team. Therefore, it causes no loss to trim theses measurements.

The following theorem describes when CBs get trimmed. Consider trimming all the state-measurement pairs of time t in the i^{th} UGV's CB. Let $k^{lj}_t(>t)$ be the first time that the l^{th} UGV communicates to the j^{th} UGV in the time interval (t,∞). Define $\tilde{k}^j_t = \max_l k^{lj}_t$, which is the time that the j^{th} UGV receives all other UGVs' measurements of t. Similarly, let $k^{ji}_t(>\tilde{k}^j_t)$ be the first time that the j^{th} UGV communicates to the i^{th} UGV in the time interval (k^j_t,∞) and define $\tilde{k}^i_t = \max_j k^{ji}_t$. The following theorem gives the bounds on when the i^{th} UGV $(\forall i \in V)$ trims all state-measurement pairs of time t in its own CB.

Theorem 3. *The i^{th} UGV trims $\left\{\left[x^l_t, z^l_t\right] (\forall l \in V)\right\}$ from its CB at the time no later than $\tilde{k}^i_t$.*

Proof. We consider the case when the lower bound is achieved. Assume that at time t after the updating step, each UGV's CB only contains the state-measurement pairs of time t. Then at $\tilde{k}_j$, the j^{th} UGV's CB has received all other UGVs' measurements at t. The i^{th} UGV can trim $\left\{\left[x^l_t, z^l_t\right] (\forall l \in V)\right\}$ only when it communicates once with all other UGVs. This

[6] '$\vee$' is the notation of the logical 'OR' operator.

happens at $\tilde{k}_t^i$ and is thus the lower bound on when the trim occurs.

Corollary 2. *The first time to trim the i^{th} UGV's CB occurs at $\tilde{k}_1^i$.*

Proof. Notice that the initial stored PDF of each UGV corresponds to the time 0. Therefore, the first time a TL is all 1's the trim occurs.

Under the frequently jointly strongly connectedness condition, it can be derived that the size of each UGV's CB is upper bounded.

Theorem 4. *The maximum interval between two consecutive trims is no greater than $(N-1)T_u$.*

Proof. See the Appendix.

Theorem 5. *The length of the i^{th} UGV's CB is no greater than NT_u.*

Proof. See the Appendix. Consider the time intervals $[k_m, k_{m+1})$, $m = 1,2,\dots$ that are frequently jointly strongly connected, where $k_1 = 0$. Define $\Delta_m = k_{m+1} - k_m$, $m = 1,2,\dots$. It is easy to know that $\Delta_i \leq T_u$. At k_2, the worst-case size of a UGV's CB is $(N-1)\Delta_1$. This can occur if UGVs do not communicate with each other until at k_2, when a UGV communicates with all others except one. At k_3, according to the definition of a frequently jointly strongly connected network, every UGV has received measurements made at or before time k_2 from all UGVs. The worst-case size of a UGV's CB is then $max\left[(N-1)(\Delta_2-\Delta_1), N(\Delta_1-\Delta_2)\right]$. The first term is the worst-case size if $\Delta_2 \geq \Delta_1$ and can be achieved similarly to the case of the time interval $[k_1, k_2)$. The second term corresponds to the worst-case size when $\Delta_1 > \Delta_2$ and can be achieved if a UGV can receive measurements from all other UGVs in the whole time interval $[k_2, k_3)$. In general, at k_{m+1}, each UGV has already received all UGVs' measurements made at or before time k_m. The worst-case size of a UGV's CB within the time interval $[k_m, k_{m+1})$ is $max\left[(N-1)(\Delta_m-\Delta_{m-1}), N(\Delta_{m-1}-\Delta_m)\right]$. Since $\Delta_m \leq T_u$, $m = 1,2,\dots$, we know that the maximum CB size of each UGV is no greater than NT_u.

4.3 Complexity of FIFO-DBF

Compared to statistics dissemination, FIFO is usually more communication-efficient for distributed filtering. To be specific, consider a grid representation of the environment with the size $D \times D$. The transmitted data between each pair of UGVs are the CB and TL of each UGV. The size of the CB is upper bounded by $O(NT_u)$, according to Theorem 5, and the size of the TL is $O(N^2)$. Therefore, the overall communication complexity is $O(N^2 + NT_u)$. On the contrary, the communicated data of a statistics dissemination approach that transmits unparameterized posterior distributions or likelihood functions is $O(D^2)$. In applications such as the target localization, D is generally much larger than N and the consensus filters usually requires multiple rounds to arrive at consensual results. Therefore, when T_u is not comparable to D^2, the FIFO protocol requires much less communication burden.

It is worth noting that each UGV needs to store an individual PDF and a sotred PDF, each of which has size $O(D^2)$. In addition, each UGV needs to keep the CB and TL. This is generally larger than that of statistics dissemination-based methods, which only stores the individual PDF. Therefore, the FIFO-DBF sacrifices the local memory for reducing the communication burden. This is actually desirable for real applications as local memory of vehicles is usually abundant compared to the limited bandwidth for communication.

Remark 3. *Under certain interaction topologies, CBs can grow to undesirable sizes and causes excessive communication burden if the trim cannot happen frequently. In this case, we can use a time window to constrain the measurements that are saved in CBs. This will cause information loss to the measurements. However, with a decently long time window, FIFO-DBF can still effectively estimate the target position.*

5 Proof of Consistency

This section proves the consistency of the maximum a posteriori (MAP) estimator of LIFO-DBF under unbiased sensors (sensors without offset). A state estimator is *consistent* if it converges in probability to the true value of the state [29]. Consistency is an important metric for stochastic filtering approaches [8] and it differs from the concept of consensus; the consensus implies that all UGVs' estimation results converge to a same value, while the consistency not only implies achieving consensus asymptotically, but also requires the estimated value converge to the true value. We first prove the consistency for static UGVs and then for moving UGVs. For simplicity and clarity, we assume that S is a finite set (e.g. a finely discretized field) and the target is static. This makes sense if the convergence is achieved much faster than the target's dynamics. We also assume that the target position can be uniquely defined by the multi-UGV network (e.g. via appropriate placement).

5.1 Static UGVs

The consistency of FIFO-DBF for static UGVs is stated as follows:

Theorem 6. *Assume the UGVs are static and the sensors are unbiased. If the network of N UGVs is frequently jointly strongly connected, then the MAP estimator of target position converges in probability to the true position of the target using FIFO-DBF, i.e.,*

$$\lim_{k\to\infty} P(X_k^{MAP} = x^g) = 1,\ i \in V,$$

where

$$X_k^{MAP} = \arg\max_X P_{pdf}^i(X|\mathbf{z}_{1:k}^i).$$

Proof. Define the time set of i^{th} UGV, $\mathcal{K}_k^{i,j}$ $(j \in V)$, that contains the time steps of measurements by the j^{th} UGV that are contained in B_k^i. The batch form of DBF at k^{th} step is

$$P_{pdf}^i(X|\mathbf{z}_{1:k}^i) = \frac{P_{pdf}^i(X) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}_k^{i,j}} P(z_t^j|X)}{\sum_{X \in S} P_{pdf}^i(X) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}_k^{i,j}} P(z_t^j|X)}.$$

Comparing $P_{pdf}^i(X_k = x|\mathbf{z}_{1:k}^i)$ with $P_{pdf}^i(X_k = x^g|\mathbf{z}_{1:k}^i)$[7] yields

$$\frac{P_{pdf}^i(x|\mathbf{z}_{1:k}^i)}{P_{pdf}^i(x^g|\mathbf{z}_{1:k}^i)} = \frac{P_{pdf}^i(x) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}_k^{i,j}} P(z_t^j|x)}{P_{pdf}^i(x^g) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}_k^{i,j}} P(z_t^j|x^g)}. \tag{6}$$

Take the logarithm of Eq. (6) and average it over k steps:

$$\frac{1}{k} \ln \frac{P_{pdf}^i(x|\mathbf{z}_{1:k}^i)}{P_{pdf}^i(x^g|\mathbf{z}_{1:k}^i)} = \frac{1}{k} \ln \frac{P_{pdf}^i(x)}{P_{pdf}^i(x^g)} + \sum_{j=1}^{N} \frac{1}{k} \sum_{t \in \mathcal{K}_k^{i,j}} \ln \frac{P(z_t^j|x)}{P(z_t^j|x^g)}. \tag{7}$$

Since $P_{pdf}^i(x)$ and $P_{pdf}^i(x^g)$ are bounded and nonzero by the choice of the initial PDF, $\lim_{k\to\infty} \frac{1}{k} \ln \frac{P_{pdf}^i(x)}{P_{pdf}^i(x^g)} = 0$. The law of large numbers yields

$$\frac{1}{k} \sum_{t \in \mathcal{K}_k^{i,j}} \ln \frac{P(z_t^j|x)}{P(z_t^j|x^g)} \xrightarrow{P} \mathbb{E}_{z_t^j} \left[\frac{P(z_t^j|x)}{P(z_t^j|x^g)} \right] \tag{8a}$$

$$= \int_{z_t^j} P(z_t^j|x^g) \frac{P(z_t^j|x)}{P(z_t^j|x^g)} dz_t^j \tag{8b}$$

$$= -D_{KL}\left(P(z_t^j|x) \| P(z_t^j|x^g) \right), \tag{8c}$$

where "$\xrightarrow{P}$" represents "convergence in probability" and $D_{KL}(P_1 \| P_2)$ denotes the Kullback-Leibler (KL) divergence between two probability distribution P_1 and P_2. KL divergence has the property that $\forall P_1, P_2,\ D_{KL}(P_1\|P_2) \leq 0$, and the equality holds if and only if $P_1 = P_2$. Therefore

$$\lim_{k\to\infty} \frac{1}{k} \sum_{t \in \mathcal{K}_k^{i,j}} \ln \frac{P(z_t^j|x)}{P(z_t^j|x^g)} < 0, \quad x \neq x^g$$

$$\lim_{k\to\infty} \frac{1}{k} \sum_{t \in \mathcal{K}_k^{i,j}} \ln \frac{P(z_t^j|x)}{P(z_t^j|x^g)} = 0, \quad x = x^g.$$

Considering the limiting case of Eq. (7), we get

$$\lim_{k\to\infty} \frac{1}{k} \ln \frac{P_{pdf}^i(x|\mathbf{z}_{1:k}^i)}{P_{pdf}^i(x^g|\mathbf{z}_{1:k}^i)} < 0, \quad x \neq x^g \tag{9}$$

$$\lim_{k\to\infty} \frac{1}{k} \ln \frac{P_{pdf}^i(x|\mathbf{z}_{1:k}^i)}{P_{pdf}^i(x^g|\mathbf{z}_{1:k}^i)} = 0, \quad x = x^g. \tag{10}$$

Eq. (9) and (10) imply that

$$\frac{P_{pdf}^i(x|\mathbf{z}_{1:k}^i)}{P_{pdf}^i(x^g|\mathbf{z}_{1:k}^i)} \xrightarrow{P} \begin{cases} 0 & x \neq x^g, \\ 1 & x = x^g. \end{cases}$$

Therefore,

$$\lim_{k\to\infty} P(X_k^{MAP} = x^g) = 1.$$

[7]For the purpose of simplicity, we use $P_{pdf}^i(x|\mathbf{z}_{1:k}^i)$ to represent $P_{pdf}^i(X_k = x|\mathbf{z}_{1:k}^i)$ in this proof.

5.2 Moving UGVs

The consistency proof for the moving UGVs case is different from the static UGVs case in that each moving UGV makes measurements at multiple different positions. We classify UGV measurement positions into two disjoint sets: *infinite-measurement spots* that contain positions where a UGV keeps revisiting as time tends to infinity, and *finite-measurement spots* that contain positions where the UGV visits finitely many times (i.e., the UGV does not visit again after a finite time period). It is easy to know that each UGV has at least one position where it revisits infinitely many times as k tends to infinity.

Theorem 7. *Assume UGVs move within a collection of finite positions and sensors are unbiased, then the MAP estimator of target position converges in probability to the true position of the target using FIFO-DBF, i.e.,*

$$\lim_{k\to\infty} P(X_k^{MAP} = x^g) = 1,\ i \in V.$$

Proof. Similar to Eq. (6), comparing $P_{pdf}^i(x|\mathbf{z}_k^i)$ and $P_{pdf}^i(x^g|\mathbf{z}_k^i)$ yields

$$\frac{P_{pdf}^i(x|\mathbf{z}_k^i)}{P_{pdf}^i(x^g|\mathbf{z}_k^i)} = \frac{P_{pdf}^i(x) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}_k^{i,j}} P(z_t^j|x; x_t^j)}{P_{pdf}^i(x^g) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}_k^{i,j}} P(z_t^j|x^g; x_t^j)}. \tag{11}$$

The only difference from Eq. (6) is that $P(z_l^j|x; x_l^j)$ in Eq. (11) varies as the UGV moves. For the finite-measurement spots, by referring to Eq. (8), it is easy to know that their contribution to Eq. (7) diminishes when $k \to \infty$. Therefore, proof using Eq. (11) can be reduced to only considering the infinite-measurement spots and the rest of the proof is similar to that of Theorem 6.

6 Simulation

We conduct a simulation that uses a team of six UGVs to localize three moving targets. Every UGV maintains three individual PDFs, each corresponding to a target. At each time step, a UGV's sensor can measurement the positions of three targets. We assume that the UGVs know the association between the measurement and the corresponding target. The targets have different motion models, including the linear motion (target 1), sinusoidal motion (target 2), and circular motion (target 3). Three of the UGVs have range-only sensors and the other three UGVs have bearing-only sensors. The interaction topology of the UGVs is time-varying and consists of four types, as shown in Fig. 3(a)-(d). A randomly generated sequence of topologies is used (Fig. 3f). It can be noticed that, the interaction topology is frequently jointly strongly connectedwhen all four types appears repeatedly (Fig. 3e). Ten layouts of the initial positions of UGVs and targets are randomly generated. We compare FIFO-DBF with CbDF and CF.

Fig. 4 show the simulation results of a specific layout. The sum of the 1st UGV's individual PDFs are shown in the

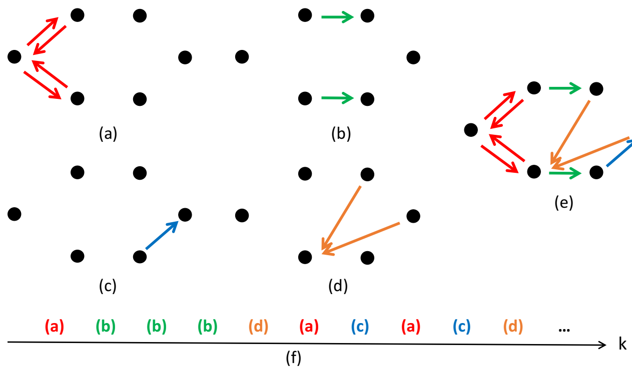


Fig. 3: The dynamically changing topologies used in the simulation: (a)-(d) four types of topologies; (e) the union of these topologies is jointly strongly connected; (f) a randomly generated sequence of topologies that satisfy the frequently jointly strongly connectedness condition.

figures. Figs. 4a to 4f show that the FIFO-DBF can successfully localize and track moving target's positions and effectively reduce the estimation uncertainty, which is similar to the performance of the CF (Fig. 4h). On the contrary, CbDF is less effective to reduce the estimation uncertainty (Fig. 4g).

We compares the three filters in terms of the estimation error and entropy of the uncertainty. The estimation error is defined as the difference between the true target position and the MAP estimate of the individual PDF:

$$\Delta_k = \|x_k^{\text{MAP}} - x_k^g\|_2.$$

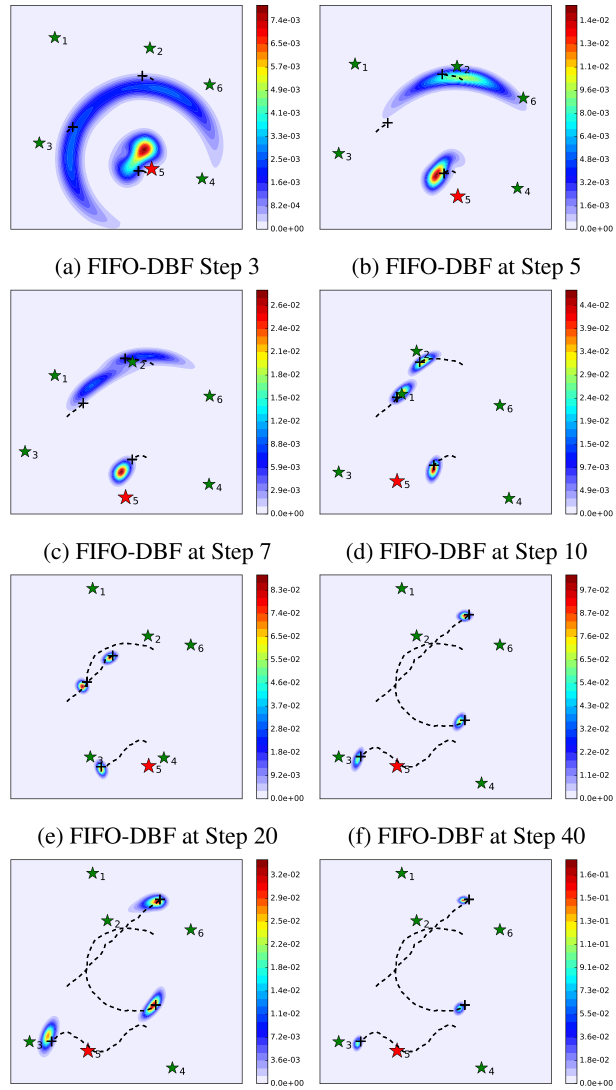
The entropy of the uncertainty is

$$H_k = \sum_{X_k \in S} -P_{pdf}(X_k)\log(P_{pdf}(X_k)).$$

The average of the estimation error and entropy of each target across ten layouts are shown in Fig. 5. It can be noticed that, the CF achieves the most accurate position estimation and fastest entropy reduction. This is an expected result since the CF utilizes all sensor measurements. The FIFO-DBF achieves similar results as the CF asymptotically. This is a very interesting results, since FIFO-DBF only communicates with neighboring UGVs and have a subset of other UGVs' measurements. The CbDF achieves similar position estimation performance and the CF and FIFO-DBF. However, it fails to effectively reduce the estimation entropy. This is because that, the linear combination of PDFs used in the CbDF does not follow the nonlinear nature of Bayesian filtering, thus information is loss during the combination. The FIFO-DBF, on the other hand, rigorously follows the procedure of Bayesian filtering, and therefore achieves better performance.

7 Conclusion

This paper presents a general measurement dissemination-based distributed Bayesian filter (DBF) method for a network of multiple unmanned ground vehicles (UGVs) under dynamically changing interaction topologies. The information exchange among UGVs relies on the Full-In-and-Full-Out (FIFO) protocol, under which UGVs exchange the communication buffers and track lists with neighbors. Under the condition that the union of the switching topologies is frequently jointly strongly connected, FIFO can disseminate measurements over the network within finite time. By using the track list, the CBs can be trimmed without causing information loss. The FIFO-DBF algorithm is then derived to estimate individual probability density function for target localization. The FIFO-DBF can significantly reduce the transmission burden between each pair of UGVs compared to the statistics



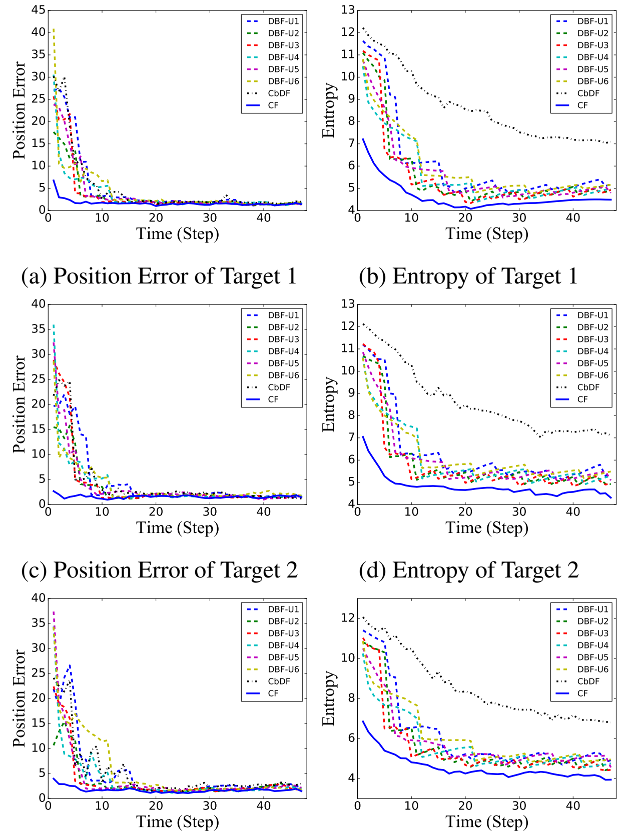
(a) FIFO-DBF Step 3 (b) FIFO-DBF at Step 5

(c) FIFO-DBF at Step 7 (d) FIFO-DBF at Step 10

(e) FIFO-DBF at Step 20 (f) FIFO-DBF at Step 40

(g) CbDF at Step 40 (h) CF at Step 40

Fig. 4: First scenario: (a) two types of topologies; (b) individual PDF of the 3rd UGV after initial observation; (c)-(e) PDFs at the end of simulation using different filters; (f) average position estimation errors; (g) average entropy of PDF. In last two figures, metrics are based on the PDFs of the 1st, 3rd and 5th UGV using FIFO-DBF, the common PDF using CbDF and using CF.



(a) Position Error of Target 1 (b) Entropy of Target 1

(c) Position Error of Target 2 (d) Entropy of Target 2

(e) Position Error of Target 3 (f) Entropy of Target 3

Fig. 5: First scenario: (a) two types of topologies; (b) individual PDF of the 3rd UGV after initial observation; (c)-(e) PDFs at the end of simulation using different filters; (f) average position estimation errors; (g) average entropy of PDF. In last two figures, metrics are based on the PDFs of the 1st, 3rd and 5th UGV using FIFO-DBF, the common PDF using CbDF and using CF.

dissemination methods. Simulations comparing FIFO-DBF with consensus-based distributed filters (CbDF) and the centralized filter (CF) show that FIFO-DBF achieves similar performance as the CF and superior performance over the CbDF while requiring less communication resource.

In our future work, we will modify the FIFO to handle the changing number of UGVs. We will also look into the combination of transmission of measurements and parameterized statistics to save communication cost without sacrificing estimation accuracy.

Appendix: Proof of the Maximal Time Interval between Consecutive Trims

(**TODO:** check if I can define my network using this definition, not the one I used before.)

Consider the time intervals $[T_m, T_{m+1})$, $m = 1, 2, \ldots$ that are frequently jointly strongly connected, where $T_1 = 0$. Within each interval, there exists at least one path between any two pairs of UGVs. Note that, $[T_m, T_{m+1})$ is different from $[k_n, k_{n+1})$. Define $L_m^{ij}\,(i, j \in V, m = 1, 2, \ldots)$ as the size of the measurements from the j^{th} UGV that are stored in the i^{th} UGV's CB at time T_m. Let C_m^i be the length of time steps of which the i^{th} UGV CB has got the measurements from all UGVs but have not yet been trimmed at T_m. Define $\Delta_m = T_{m+1} - T_m$. Therefore $\Delta_m \leq T_M$. It is easy to know that $L_1^{ij} \leq T_M$. The equality holds if the i^{th} UGV does not receive the measurement of the j^{th} UGV until at T_2. We consider the upper bound of L_m^{ij}, $m = 1, 2, \ldots$. Notices that, if $\Delta_m > C_m^i$, then it can happen that after trimming all measurements corresponding to first C_m^i time steps, no trimming happens if the i^{th} UGV does not receive the measurement of the j^{th} UGV until at T_{m+1}. This case will maximally increase the size of the i^{th} UGV's CB. Considering the mechanism of the trimming, we can derive maximal increase of the CB's size:

$$
\begin{aligned}
L_{m+1}^{ij} &= L_m^{ij} + \max(\Delta_m - C_m^i, 0), \; m = 1, 2, \ldots \\
C_{m+1}^i &= C_m^i + \Delta_m - \min(C_m^i, \Delta_m) = \max(\Delta_m, C_m^i).
\end{aligned}
$$

We consider the relation of between L_{m+3}^{ij} and L_m^{ij}:

$$
\begin{aligned}
L_{m+3}^{ij} &= L_{m+2}^{ij} + \max(\Delta_{m+2} - C_{m+2}^i, 0) && \text{(12a)} \\
&= L_{m+1}^{ij} + \max(\Delta_{m+1} - C_{m+1}^i, 0) + \max(\Delta_{m+2} - C_{m+2}^i, 0) && \text{(12b)} \\
&= L_{m+1}^{ij} + \max(\Delta_{m+1} - C_{m+1}^i, 0) + \max(\Delta_{m+2} - \max(\Delta_{m+1}, C_{m+1}^i), 0) && \text{(12c)} \\
&= L_{m+1}^{ij} + \max(\Delta_{m+2}, \max(\Delta_{m+1}, C_{m+1}^i)) - C_{m+1}^i && \text{(12d)} \\
&= L_m^{ij} + \max(\Delta_m - C_m^i, 0) + \max(\Delta_{m+2}, \max(\Delta_{m+1}, C_{m+1}^i)) - C_{m+1}^i && \text{(12e)} \\
&= L_m^{ij} + \max(\Delta_m - C_m^i, 0) + \max(\Delta_{m+2}, \max(\Delta_{m+1}, \max(\Delta_m, C_m^i))) \\
&\quad - \max(\Delta_m, C_m^i) && \text{(12f)} \\
&= L_m^{ij} + \max(\Delta_{m+2}, \max(\Delta_{m+1}, \max(\Delta_m, C_m^i))) - C_m^i. && \text{(12g)}
\end{aligned}
$$

Eq. (12a) and (12b) are from the definitions of L_{m+2}^{ij} and L_{m+1}^{ij}, and Eq. (12c) uses the definition of C_{m+2}^i. Eq. (12d) is obtained by using the following relation:

$$
\max(a, b) = \max(a - b, 0) + b, \quad \forall a, b \in \mathbb{R}.
$$

Eq. (12e) to (12g) are similarly derived. In general, we can get the following result for $m = 1, 2, \ldots$:

$$
L_m^{ij} = L_1^{ij} + \max(\Delta_{m-1}, \max(\Delta_{m-2}, \max(\Delta_{m-3}, \ldots, \max(\Delta_1, C_1^i))) - C_1^i.
$$

Notice that $\Delta_m \leq T_M$, $L_1^{ij} \leq T_M$ and $C_1^i = 0$. Therefore, $L_m^{ij} \leq 2T_M$.

References

[1] J.-F. Chamberland and V. V. Veeravalli, "Wireless sensors in distributed detection applications," *Signal Processing Magazine, IEEE*, vol. 24, no. 3, pp. 16–25, 2007.

[2] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 842–849.

[3] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 8179–8184.

[4] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed cooperative outdoor multirobot localization and mapping," *Autonomous Robots*, vol. 17, no. 1, pp. 23–39, 2004.

[5] D. Gu, "Distributed particle filter for target tracking," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3856–3861.

[6] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using bayesian consensus filtering," in *American Control Conference (ACC), 2014*, pp. 634–641.

[7] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.

[8] Z. Chen, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[9] L. Zuo, K. Mehrotra, P. K. Varshney, and C. K. Mohan, "Bandwidth-efficient target tracking in distributed sensor networks using particle filters," in *Information Fusion, 2006 9th International Conference on*, pp. 1–4.

[10] M. Vemula, M. F. Bugallo, and P. M. Djurić, "Target tracking in a two-tiered hierarchical sensor network," in *ICASSP 2006 Proceedings.*, vol. 4, pp. IV–IV.

[11] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, vol. 48, no. 6, pp. 988–1001, 2003.

[12] W. Ren, R. W. Beard *et al.*, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on automatic control*, vol. 50, no. 5, pp. 655–661, 2005.

[13] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

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

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

[16] J. Beaudeau, M. F. Bugallo, and P. M. Djuric, "Target tracking with asynchronous measurements by a network of distributed mobile agents," in *in ICASSP 2012 Proceedings*, pp. 3857–3860.

[17] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 24.

[18] M. Coates, "Distributed particle filters for sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 99–107.

[19] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks-part ii: unknown probability density function," *Signal Processing, IEEE Transactions on*, vol. 54, no. 7, pp. 2784–2796, 2006.

[20] P. M. Djurić, J. Beaudeau, and M. F. Bugallo, "Non-centralized target tracking with mobile agents," in *ICASSP 2011 Proceedings*, pp. 5928–5931.

[21] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *Automatic Control, IEEE Transactions on*, vol. 53, no. 8, pp. 1804–1816, 2008.

[22] Y. Xia, J. Shang, J. Chen, and G.-P. Liu, "Networked data fusion with packet losses and variable delays," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 5, pp. 1107–1120, 2009.

[23] K. Y. Leung, T. D. Barfoot, and H. H. Liu, "Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 62–77, 2010.

[24] C. Liu, S. E. Li, and J. K. Hedrick, "Distributed target localization using a group of ugvs under dynamically changing interaction topologies," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 2585–2590.

[25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[26] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[27] A. N. Bishop, B. Fidan, B. D. Anderson, K. Doğançay, and P. N. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.

[28] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a bayesian world," in *Field and service robotics*. Springer, 2003, pp. 209–222.

[29] T. Amemiya, *Advanced econometrics*. Harvard university press, 1985.