Estimation of Moving Targets Using Distributed Bayesian Filter Under Dynamically Changing Networks

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 distributed Bayesian filtering (DBF) method for a network of multiple unmanned ground vehicles (UGVs) under dynamically changing interaction topologies. The information exchange among UGVs relies on a measurement dissemination scheme, called Latest-In-and-Full-Out (FIFO) protocol. Different from statistics dissemination approaches that transmit posterior distributions or likelihood functions, each UGV under FIFOonly sends a buffer that contains latest available measurements to neighboring nodes, which significantly reduces the transmission burden between each pair of UGVs to scale linearly with the size of the network. Under the condition that the union of undirected switching topologies is connected frequently enough, FIFOcan disseminate observations over the network within finite time. The FIFO-based DBF algorithm is then derived to estimate individual probability density function (PDF) for target localization in a static environment. The consistency of this algorithm is proved that each individual estimate of target position converges in probability to the true target position. The effectiveness of this method is demonstrated by comparing with consensus-based distributed filters and the centralized filter in simulations.

1 INTRODUCTION

Unmanned ground vehicles (UGV) that operate without on-board operators have been used for many applications that are inconvenient, dangerous, or impossible to human. Distributed estimation using a group of networked UGVs has been applied to collectively infer status of complex environment, such as intruder detection [1] and object tracking [2]. Several techniques have been developed for distributed estimation, including distributed linear Kalman filters (DKF) [3], distributed extended Kalman filters [4] and distributed particle filters [5], etc. The most generic filtering scheme is distributed Bayesian filters (DBF), which can be applied for nonlinear systems with arbitrary noise distributions [6,7]. This paper focuses on a communication-efficient DBF for networked UGVs.

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 [8,9]. In the latter, each agent individually executes distributed estimation and the agreement of local estimates is achieved by certain consensus strategies [10–12]. 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 [13]. In the MD scheme, instead of exchanging statistics, each agent sends its observations 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 distributed linear Kalman filter (DKF) for estimating states of

[*]Address all correspondence to this author.

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 [14]. 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 [15].

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 [16], 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 [17]. Ribeiro et al. (2006) exchanged quantized observations along with error-variance limits considering more pragmatic signal models [18]. 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 [19]. 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 [20]. In such cases, dynamically changing topologies can cause random packet loss and variable transmission delay, thus decreasing the performance of distributed estimation, and even leading to inconsistency and non-consensus. Leung et al. (2010) explored a decentralized filter for dynamic robot networks [21]. 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, 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. In this work, we introduce the concept of the track list and extend our methods to time-varying topologies. 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. Individual Bayesian filter is implemented locally by each UGV after exchanging observations using FIFO. Under the condition that the union of undirected switching topologies is connected frequently enough, two properties are achieved: (1) FIFO can disseminate observations over the network within finite time; (2) FIFO-based DBF guarantees the consistency of estimation that each individual estimate of target position converges in probability to the true target position as the number of observations tends to infinity. The main benefit of using FIFOis on the reduction of communication burden, with the transmission data volume scaling linearly with the size of the UGV network.

The rest of this paper is organized as follows: the FIFO protocol for dynamically changing interaction topologies is formulated in **??**; the FIFO-based DBF algorithm is described in Section 4, where the consistency of estimation is proved; 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 environmental perception. Due to the limit of communication range, each UGV can only exchange sensor observations with its neighbors. The Bayesian filter is run locally on each UGV based on its own and received observations via single-hopping to estimate the position of a static target in S.

2.1 Target and Sensor Model

The target motion takes a deterministic discrete-time model that can be described by

$$x_{k+1}^{g} = f(x_k^g), \tag{1}$$

where $x_k^g \in \mathbb{R}^2$ denotes the target position at time k; u_k^g represents the control input of the target.

Each UGV constantly measures the target position and the sensor measurement of i^{th} UGV can be described by a stochastic model:

$$z_k^i = g_i(x_k^g, w_k^i; y_k^i), \tag{2}$$

where w_k^i is the white measurement noise and $y_k^i = [x_k^i; \theta_k^i]$ represents the sensor state, consisting of the sensor position x_k^i and direction θ_k^i. g_i depends on the type of the sensor. Let $\mathcal{F}(y_k^i)$ denote the sensor field of view (FOV), g_i for several typical sensor types can be defined as follows:

Range-only sensors: when the target is within the sensor's FOV, the measurement only depends on the relative distance between the sensor and the target.

$$g_i(x_k^g, w_k^i; x_k^i) = \begin{cases} \|x_k^g - x_k^i\|_2 + w_k^i & \text{if} x_k^g \in \mathcal{F}(y_k^i), \\ \emptyset & \text{if} x_k^g \notin \mathcal{F}(y_k^i). \end{cases} \quad (3)$$

Bearing-only sensors: when the target is within the sensor's FOV, the measurement only depends on the relative bearing between the sensor and the target.

$$g_i(x_k^g, w_k^i; x_k^i) = \begin{cases} \measuredangle(x_k^g - x_k^i) + w_k^i & \text{if} x_k^g \in \mathcal{F}(y_k^i), \\ \emptyset & \text{if} x_k^g \notin \mathcal{F}(y_k^i). \end{cases} \quad (4)$$

A probabilistic sensor model that describes the conditional probability of a certain measurement given sensor and target state is a key component for Bayesian filtering. We define a likelihood function to represent the probability of the target being detected by a sensor:

$$p_{1,k}^i = P(z_k^i \neq \emptyset | x_k^g; x_k^i) \in [0,1], \, x_k^g \in S, \quad (5)$$

where x_k^i is the i^{th} sensor's position. Correspondingly, the likelihood function for no target being detected is:

$$p_{0,k}^i = P(z_k^i = \emptyset | x_k^g; x_k^i) = 1 - p_{1,k}^i. \quad (6)$$

The combination of Eq. (5) and Eq. (6) forms the probabilistic model for a sensor. If w_k^i is a zero-mean Gaussian white noise, then the probabilistic sensor model can be described as

$$\begin{cases} p_{1,k}^i \sim \mathcal{N}(\bar{z}_k^i, \Sigma_k^i) & \text{if} x_k^g \in \mathcal{F}(y_k^i) \\ p_{1,k}^i = 0 & \text{if} x_k^g \in \mathcal{F}(y_k^i), \end{cases} \quad (7)$$

where $\bar{z}_k^i$ is the nominal value of the measurement and equals $\|x_k^g - x_k^i\|$ and $\measuredangle(x_k^g - x_k^i)$ for range-only and bearing-only sensors, correspondingly.

Consequently,

$$p_{0,k}^i = \begin{cases} 0 & \text{if} x_k^g \in \mathcal{F}(y_k^i) \\ 1 & \text{if} x_k^g \in \mathcal{F}(y_k^i). \end{cases} \quad (8)$$

For the purpose of simplicity, we will not explicitly write x_k^i in the sensor model (Eq. (5) and (6)) for the rest of the paper.

Remark 1. *Given the knowledge of current target and UGV positions, current observation by each UGV can be considered conditionally independent from its own past observations and those by other UGVs [22].*

Remark 2. *The proposed FIFO protocol and the consistency property to be described in Section 4 are applicable for general sensors, not limited to the ones described in this section. In addition, they do not rely on the Gaussian noise assumption.*

2.2 Graphical Model of Interaction Topology

We consider a simple[1] graph $G = (V, E)$ to represent the interaction topology of N networked UGVs, where the vertex set $V = \{1, \ldots, N\}$ represents the index collection 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. 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 a communication link exists from the i^{th} to j^{th} UGV and $m_{ij} = 0$ indicates no communication from i to j. A directed graph is *strongly connected* if there is a directed path connecting any two arbitrary vertices in V[2].

Let $\bar{G}$ denote the set of all possible simple directed graphs[3] defined over the network of UGVs. It is easy to know that $\bar{G}$ has finite elements. The adjacency matrix associated with a graph $G_l \in \bar{G}$ is denoted as $A_l = [a_{ij}^l]$. Define the *union* of a collection of graphs $\{G_{i_1}, G_{i_2}, \ldots, G_{i_l}\} \subset \bar{G}$ as the graph with the nodes in V and the edge set given by the union of edge sets of G_{i_j}, $j = 1 \ldots, l$. Such collection is defined to be *jointly strongly connected* if the union of its members forms a strongly connected graph [4].

We define two concepts of neighbors in a UGV network. The *direct neighbors* of i^{th} UGV under topology G_l is defined as $\mathcal{N}_i(G_l) = \{ j | a_{ij}^l = 1, \, j \in V \}$. All UGVs in $\mathcal{N}_i(G_l)$ can directly receive information from the i^{th} UGV via single-hopping. The *accessible neighbors* is defined as $\mathcal{Q}_i(G_l)$ that contains indices of UGVs whose information can be received by the i^{th} UGV, possibly via multi-hopping, given a specific data exchange protocol and the interaction topology G_l.

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

This study proposes a Full-In-and-Full-Out (FIFO) protocol for observation exchange. In our previous work, we proposed a Latest-In-and-Full-Out (LIFO) protocol and can be used for time-invariant topologies. FIFO is suitable for time-varying topologies. Let $Y_{\mathcal{K}}^i = \{ [x_k^i, z_k^i] | k \in \mathcal{K} \}$ be the set of state-measurement pairs of robot i, where $\mathcal{K}$ is an index set of time steps. Under FIFO, each UGV contains a communication buffer (CB) to store a subset of measurements and the corresponding states of all UGVs:

$$B_k^i = \left[Y_{\mathcal{K}_k^{i,1}}^1, \ldots, Y_{\mathcal{K}_k^{i,N}}^N \right],$$

[1] 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).

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

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

[4] For undirected graphs, such collection is jointly connected [10]

where B_k^i is the CB of i^{th} robot at time k and $Y^j_{\mathcal{K}_k^{i,j}}$ represents the set of robot j's measurements at time steps in $\mathcal{K}_k^{i,j}$ that are stored in robot i's CB. Note that under FIFO and certain conditions (Proposition 1) of interaction topologies, $\mathcal{Q}_i = \{1,\dots,N\} \setminus \{i\}$, i.e. each robot can know the measurements from all other robots. This will be proved in Corollary 1. Let $G_k \in \bar{G}$ represent the interaction topology at time k. The **FIFO protocol** is stated in Algorithm 1. Note that in the Updating Step, the algorithm uses Algorithm 3, which we will introduce in Section 4.2. For the purpose of clarity, we ignore this operation and do not trim CB at this stage.

Fig. 1 illustrates the FIFO cycles of a network of 3 UGVs with switching line topologies. There are two types of topologies: under the first one only UGV 1 and UGV 2 can directly communicate and under second one only UGV 2 and UGV 3 can directly communicate. Several facts can be noticed in Fig. 1: (1) the two topologies are jointly connected within each time intervals $[0,3)\,,[3,5)\,,[5,7)$; (2) (**TODO:** may need to change) CBs of all UGVs are filled within 5 steps; (3) after being filled, each CB keeps updated every finite time steps, which means each UGV receives new observations of other UGVs with finite delay. Extending these facts to a network of N UGVs, we have the following proposition:

(**TODO:** it seems to me that a tighter lower bound can be achieved by using FIFO, but not sure how to compute it.)

Proposition 1. *Consider a network of N UGVs with switching interaction topologies. If the following two conditions are satisfied:*

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 connected,*

then arbitrary 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 $T_u = \sup\limits_{m=1,2,\dots} (k_{m+1} - k_m)T$ is the upper bound of interval lengths.

Proof. Without loss of generality, we consider the transmission of B_1^i from i^{th} UGV to an arbitrary one j. Since each robot will receive neighbors' CBs and send the merged one to its neighbors, i^{th} UGV can transmit B_1^i to j if and only if there is a path from node i to j in the interaction topology. 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 $[k_m, k_{m+1})\,,\ ,m = 2,3,\dots,$ it can be shown that all $N-1$ UGVs, besides the i^{th} UGV, will have the information in B_1^i no later by k_N. Therefore, the transmission delay between an arbitrary pair of UGVs is no greater than $(N-1)T_u$.

Algorithm 1 FIFO Protocol

```
(1) Initialization.
    CB: The CB of $i^{\text{th}}$ UGV is initialized at $k = 0$:
        $B_0^i = \left[Y^1_{\mathcal{K}_0^{i,1}}, \dots, Y^N_{\mathcal{K}_0^{i,N}}\right]$, where $Y^j_{\mathcal{K}_0^{i,j}} = \left\{\left[x_0^j, \varnothing\right]\right\}$.
    TL: The TL of $i^{\text{th}}$ UGV is initialized when $k = 0$:
        $P_0^i = \mathbf{0}$, i.e. $p_0^{j,l} = 0, \forall j,l \in \{1\dots,N\}$.
(2) At time $k\,(k \geq 1)$ for $i^{\text{th}}$ UGV:
(2.1) Receiving Step.
    CB: The $i^{\text{th}}$ UGV receives all CBs of its direct neighborhood $\mathcal{N}_i(G_{k-1})$. The received CBs are totally $|\mathcal{N}_i(G_{k-1})|$ groups, each of which corresponds to the $(k-1)^{\text{th}}$ step CB of a UGV in $\mathcal{N}_i(G_{k-1})$. The received CB from $l^{\text{th}}$ UGV is
        $B_{k-1}^l = \left[Y^1_{\mathcal{K}_{k-1}^{l,1}}, \dots, Y^N_{\mathcal{K}_{k-1}^{l,N}}\right],\ l \in \mathcal{N}_i(G_{k-1})$
    TL: The $i^{\text{th}}$ UGV receives all TLs of its direct neighborhood $\mathcal{N}_i(G_{k-1})$. The received TL from $l^{\text{th}}$ $(l \in \mathcal{N}_i(G[k-1]))$ UGV is $P_{t_l}^l$.
(2.2) Observation Step.
    CB: The $i^{\text{th}}$ UGV updates $Y^i_{\mathcal{K}_k^{i,i}}$ by its own state-measurement pair at current step:
        $Y^i_{\mathcal{K}_k^{i,i}} = Y^i_{\mathcal{K}_{k-1}^{i,i}} \cup \left\{x_k^i, z_k^i\right\}$.
(2.3) Updating Step.
    CB: The $i^{\text{th}}$ UGV updates other elements of its own CB, i.e., $Y^j_{\mathcal{K}_k^{i,j}}\ (j \neq i)$, by merging with all received CBs:
        $Y^j_{\mathcal{K}_k^{i,j}} = Y^j_{\mathcal{K}_{k-1}^{i,i}} \cup Y^j_{\mathcal{K}_{k-1}^{l,j}}, \forall \forall j \neq il \in \mathcal{N}_i(G_{k-1})$.
    TL: The $i^{\text{th}}$ UGV updates its own TL using all the received TLs:
        if $k^{i,j} > k^{i,l}$, keep current $\mathbf{p}^j_{k^{i,j}}$;
        if $k^{i,j} = k^{i,l}$, $\mathbf{p}^j_{k^{i,j}} = \mathbf{p}^j_{k^{i,j}} \vee \mathbf{p}^j_{k^{l,j}}$;        $\forall j \in \{1\dots,N\}$
        if $k^{i,j} < k^{i,l}$, $\mathbf{p}^j_{k^{i,j}} = \mathbf{p}^j_{k^{l,j}}$ and $k^{i,j} = k^{i,l}$.
    Trim the CB based on the updated track lists, see Algorithm 3.
(2.4) Sending Step:
    CB: The $i^{\text{th}}$ UGV broadcasts its updated CB to all of its neighbors defined in $\mathcal{N}_i(G_k)$.
    TL: The $i^{\text{th}}$ UGV broadcasts its updated track list to all of its neighbors defined in $\mathcal{N}_i(G_k)$.
(3) $k \leftarrow k + 1$ until stop
```

We defined the interaction topology that satisfies the two conditions in Proposition 1 as a frequently jointly strongly connected (FJSC) network.

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

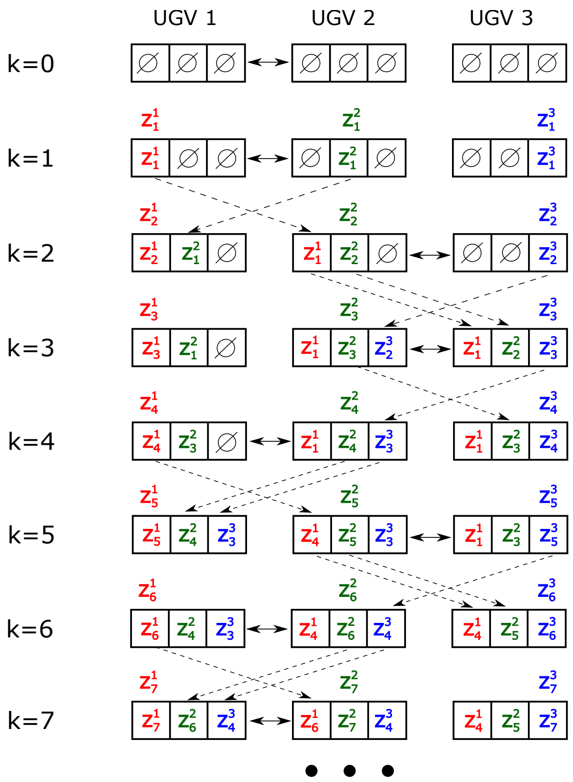

UGV 1
UGV 2
UGV 3
k=0
k=1
k=2
k=3
k=4
k=5
k=6
k=7
z_1^1 z_1^2 z_1^3
z_2^1 z_2^2 z_2^3
z_3^1 z_3^2 z_3^3
z_4^1 z_4^2 z_4^3
z_5^1 z_5^2 z_5^3
z_6^1 z_6^2 z_6^3
z_7^1 z_7^2 z_7^3
$\varnothing$ $\varnothing$ $\varnothing$ | $\varnothing$ $\varnothing$ $\varnothing$ | $\varnothing$ $\varnothing$ $\varnothing$
z_1^1 $\varnothing$ $\varnothing$ | $\varnothing$ z_1^2 $\varnothing$ | $\varnothing$ $\varnothing$ z_1^3
z_2^1 z_1^2 $\varnothing$ | z_1^1 z_2^2 $\varnothing$ | $\varnothing$ $\varnothing$ z_2^3
z_3^1 z_1^2 $\varnothing$ | z_1^1 z_3^2 z_2^3 | z_1^1 z_2^2 z_3^3
z_4^1 z_3^2 $\varnothing$ | z_1^1 z_4^2 z_3^3 | z_1^1 z_3^2 z_4^3
z_5^1 z_4^2 z_3^3 | z_4^1 z_5^2 z_3^3 | z_1^1 z_3^2 z_5^3
z_6^1 z_4^2 z_3^3 | z_4^1 z_6^2 z_4^3 | z_4^1 z_5^2 z_6^3
z_7^1 z_6^2 z_4^3 | z_6^1 z_7^2 z_4^3 | z_4^1 z_5^2 z_7^3
● ● ●

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

FIFO within finite time. This implies $Q_i = \{1,\dots,N\}\setminus\{i\}$.

Proof. According to Proposition 1, the transmission delay between an arbitrary pair of UGVs is no greater than $(N-1)T_u$. Therefore, each UGV is guaranteed to receive B^j_t, $\forall t \geq 0$, $j \in V$ when $k \geq t+(N-1)T_u$.

Remark 3. *The frequency that each UGV receive other UGVs' CBs depends on the property of the network. Corollary 1 gives an upper bound on the transmission delay time for all FIFO networks.*

4 Distributed Bayesian Filter via FIFO Protocol

The generic distributed Bayesian filter (DBF) is introduced in this section. Let $X_k \in S$ be the random variable that represents the position of the target at time k. The probability density function (PDF) of X_k, called *individual PDF*, of i^{th} UGV is then represented by $P^i_{pdf}(X_k|\mathbf{z}^i_{1:k})$, where $\mathbf{z}^i_{1:k}$ denotes the set of measurements by i^{th} UGV and by UGVs in Q_i, that have been received by i^{th} UGV until time k. The initial individual PDF, $P^i_{pdf}(X_0)$, is constructed given all available prior information including past experience and environment knowledge. It is necessary to initialize the individual PDF such that the probability density of true target position is nonzero, i.e., $P^i_{pdf}(X_0 = x^g_0) \neq 0$.

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

Algorithm 2 FIFO-DBF Algorithm

For i^{th} UGV at k^{th} step ($\forall i \in V$):
After the updating step in Algorithm 1,
(1) Initialize a *temporary PDF* by assigning the stored individual PDF to it:

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

where the stored individual PDF is for time t:

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

(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

$$P^i_{tmp}(X_\xi) = K_\xi P^{pre}_{tmp}(X_\xi)\prod_{j\in\Omega^i_\xi} P(z^j_\xi|X_\xi),$$

$$K_\xi = \left[\int_S P^{pre}_{tmp}(X_\xi)\prod_{j\in\Omega^i_\xi} P(z^j_\xi|X_\xi)dX_\xi\right]^{-1}.$$

(2.3) When $\xi = t+1$, if $z^j_{t+1} \neq \emptyset$ for $\forall j \in V$, then the stored PDF will be updated to be the temporary PDF of time $t+1$:

$$P^i_{stored,t+1} = P^i_{tmp}(X_{t+1}),$$

where

$$P^i_{tmp}(X_{t+1}) = P^i_{pdf}(X_{t+1}|z^1_{1:t+1},\dots,z^N_{1:t+1}).$$

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

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

$$P^i_{pdf}(X_k|\mathbf{z}^i_{1:k-1}) = \int\limits_{X_{k-1}\in S} P(X_k|X_{k-1})P^i_{pdf}(X_{k-1}|\mathbf{z}^i_{1:k-1})dX_{k-1}, \tag{9}$$

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

$$P(X_k = c_k|X_{k-1} = c_{k-1}) = \begin{cases} 1 & \text{if} \quad c_k = f(c_{k-1},u^g_{k-1}) \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

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

$$P^i_{pdf}(X_k|\mathbf{z}^i_{1:k}) = K_i P^i_{pdf}(X_k|\mathbf{z}^i_{1:k-1})P(\mathbf{z}^i_k|X_k), \tag{11}$$

where $P(\mathbf{z}^i_k|X_k)$ comes from the sensor model and K_i is a normalization factor, given by:

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

4.1 FIFO-DBF

This section derives the FIFO-DBF for localizing a target. At time k, each UGV maintains two PDFs, the *stored PDF* and the *individual PDF*. The stored PDF, $P^i_{sto,t}(X_t)$, is updated from the i^{th} UGV's initial PDF by fusing the state-measurement pairs of all UGVs up to a certain time $t \leq k$. Therefore, the choice of t needs to ensure that the i^{th} UGV has received the state-measurement pairs of times no greater than t from all UGVs, which is described in Section 4.2 by using the track lists. The individual PDF, $P^i_{pdf}(X_k|B^i_k)$, is then computed from $P^i_{sto,t}(X_t)$ by fusing the measurements of times from $t+1$ to k in the CB, running the Bayesian filter (Eq. (9) and (11)). Note that initially, $P^i_{stored,0} = P^i_{pdf}(X_0)$.

The **FIFO-DBF algorithm** is stated in Algorithm 2. At the beginning, we assign the stored PDF to a temporary PDF, which will then be updated based on measurements in the CB to obtain the individual PDF. We use Ω^i_ξ $(\xi = t+1, \ldots, k)$ to denote the index set of UGVs whose state-measurement pair of time (ξ) is stored in i^{th} UGV's CB, i.e. $\Omega^i_\xi = \left\{ j \in Q_i \bigcup \{i\} \,|\, \left[x^j_\xi, z^j_\xi\right] \in B^i_k \right\}$. The measurements will be sequentially used to update the temporary PDF until the latest measurements are used. The temporary PDF is then assigned as the individual PDF of time k. It should be noted that, when the UGV's CB contains all UGVs' state-measurement pairs of time $t+1$, the stored PDF will be replaced by the temporary PDF of $t+1$. Fig. 2 illustrates the FIFO-DBF procedure for the 1st 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, each UGV only needs to start from the stored PDF each time it computes the individual PDF. We point out that the times of the stored PDF of all UGVs can be different from each other. The stored PDF is saved locally by each UGV, not transmitted to others to save communication resource. [5].

4.2 Track Lists for Trimming Communication Buffers

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, while ensuring no information is lost in the sense that all UGVs' measurements have been used for update hte individual PDF. To trim the unnecessary measurements, each robot keeps a *track list* (TL), $P^i_k = \left[\mathbf{p}^1_{k^{i,1}}, \ldots, \mathbf{p}^N_{k^{i,N}}\right]^T$ $(i \in V)$, where $\mathbf{p}^j_{k^{i,j}} = \left[p^{j,l}_{k^{i,j}}, l \in V\right]$ is a binary vector with size N. For

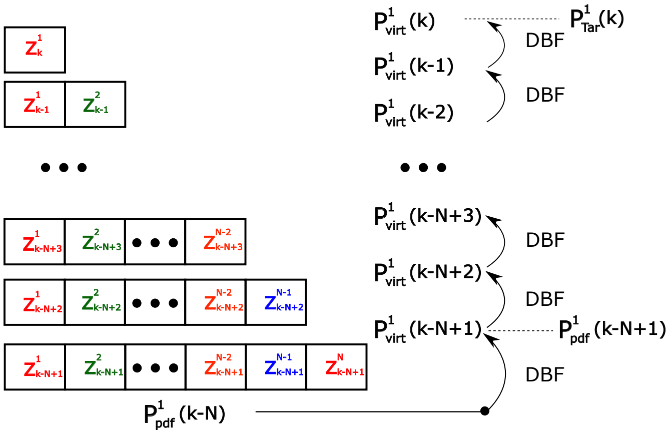


Fig. 2: Example of FIFO-DBF for 1st UGV at time k. Networked UGVs take a line topology. The stored individual PDF is represented by $P^1_{pdf}(k-N)$. The UGV first calculates $P^1_{tmp}(k-N+1)$, defined in Algorithm 2, and then stores it as $P^1_{pdf}(k-N+1)$. Repeating DBF until obtaining $P^1_{pdf}(k)$.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

the i^{th} UGV, the TL P^i_k represents its knowledge of the oldest measurements, in terms of the measurement time, by all UGVs. Each element $p^{j,l}_{k^{i,j}}$ equals 1 if the i^{th} robot knows that $\left[x^l_{k^{i,j}}, z^l_{k^{i,j}}\right]$ has been received by the j^{th} robot, and equals 0 if the i^{th} robot cannot determine whether $\left[x^l_t, z^l_t\right]$ has been received by the j^{th} robot. Mathematically speaking,

$$p^{j,l}_{k^{i,j}} = \begin{cases} 1 & \text{if} \quad \exists t \in \mathbb{N}, \text{ s.t. } k^{i,j} \leq t \leq k \text{ and } \left[x^l_{k^{i,j}}, z^l_{k^{i,j}}\right] \in B^j_t, \\ 0 & \text{if} \quad \nexists t \in \mathbb{N}, \text{ s.t. } k^{i,j} \leq t \leq k \text{ and } \left[x^l_{k^{i,j}}, z^l_{k^{i,j}}\right] \in B^j_t. \end{cases}$$

It can happen that $\left[x^l_{k^{i,j}}, z^l_{k^{i,j}}\right]$ has received by j^{th} UGV but i^{th} UGV does not know this and thus $p^{j,l}_{k^{i,j}} = 0$. When all elements of P^i_k are 1's, it means that the robot i is sure that the j^{th} UGV $(\forall j \in V)$ has received the state-measurement pairs of time $k^{i,j}$ from all UGVs. Choose the minimum time $k^i_m = \min\limits_j k^{i,j}$. Then the i^{th} robot fuse all the measurements of time no greater than k^i_m in its own CB to update the stored PDF. TLs are exchanged when UGVs communicate and are used to trim the CBs. The exchange and updating of TLs are described in the TL part in Algorithm 1 and Algorithm 3 describes the approach to trim CBs using TLs. Since a TL only keeps track of each robot's oldest measurement, CBs are only trimmed by one time step.

The use of TLs can avoid the excessive size of CBs and guarantee that fusing the measurements and 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 1. *For a frequently jointly strongly connected network using FIFO-DBF, each UGV's individual PDF is updated with measurements from all UGVs. Each UGV's es-*

[5] (**TODO:** change notations here) Due to the space limit, in this figure we use $P^i_{pdf}(k)$, $P^i_{pdf}(k-N)$ and $P^i_{pdf}(k-N+1)$ to represent $P^i_{pdf}(X|\mathbf{z}^i_{1:k})$ $P^i_{pdf}(X|\mathbf{z}^i_{1:k-N})$ and $P^i_{pdf}(X|\mathbf{z}^i_{1:k-N+1})$, respectively.

timation result with the trimmed CB by using the TL is the same as with the non-trimmed CB.

Proof. According to Corollary 1, each UGV receives the state-measurement pairs from all other UGVs within finite time steps, which are used for updating the individual PDF.

Let $k_m = \min_j k^{i,j}$. Trimming P^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, i.e., $\left[x^l_{k_m}, z^l_{k_m}\right]$, $l \in V$. A UGV has either saved the pairs in its CB or fused them to update its individual PDF. In either case, such pairs are no longer needed to be transmitted since it will not add any unused information to the team. Therefore, it causes no loss to trim theses measurements.

The following theorem describes how often CBs get trimmed. Consider trimming the state-measurement pairs of time t in i^{th} UGV's CB. Let $t_{lj}(> t)$ be the first time that a path from the l^{th} UGV to the j^{th} UGV exists in the time interval (t, ∞). Similarly, define $t_{ji}(> t_{lj})$ as the first time that a path from the j^{th} UGV to the i^{th} UGV exists in the time interval (t_{lj}, ∞). Then the following theorem gives the lower bound on when the i^{th} UGV trims all state-measurement pairs of time t in its own CB.

Theorem 2. *The i^{th} UGV trims $\left\{\left[x^l_t, z^l_t\right] \ (l \in V)\right\}$ from its CB at time k^i and $\max_{l,j,i} t_{ji} \leq k^i \leq t + 2(N-1)T_u$, where $T_u = \sup_{m=1,2,\ldots} (k_{m+1} - k_m)T$.*

Proof. The first time for the j^{th} UGV to receive B^l_t from l^{th} UGV is time t_{lj} and the j^{th} UGV's TL is updated so that $p^{j,l}_{t'} = 1$ for some $t' \leq t$. If $t' = t$, then all measurements before t from the l^{th} UGV has already been fused into $P^j_{stored,t-1}$. The first time the i^{th} UGV knows $p^{j,l}_{t'} = 1$ is at time t_{ji}, when the j^{th} UGV's TL is transmitted to i^{th} UGV. Therefore, $\max_{l,j,i} t_{ji}$ is the earliest time when all UGVs has got guaranteed about each other UGV's reception of all measurements of time t.

If i^{th} UGV's stored PDF corresponds to the time step $t-1$, then the trim happens at $k^i = \max_{l,j,i} t_{ji}$. If the stored PDF corresponds to a time step less than $t-1$, then the trim happens at $k^i > \max_{l,j,i} t_{ji}$. Therefore $k^i \geq \max_{l,j,i} t_{ji}$. The upper bound can be obtained using Proposition 1.

Corollary 2. *The first time to trim i^{th} CB occurs at $k^i = \max_{l,j,i} t_{ji}$.*

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

Corollary 3. *The time between two consecutive trims of the i^{th} UGV's CB is*

$$\Delta k_i = \begin{cases} 1 + n'_{i,t+1} - n_{i,t} & \mathit{if}\, n'_{i,t+1} \geq n_{i,t} \\ 1 & \mathit{if}\, n'_{i,t+1} < n_{i,t} \end{cases},$$

where $n_{i,t} = \max_{l,j} t_{ji}$ starting from t and $n'_{i,t+1} = \max_{l,j} t_{ji}$ starting from $t+1$.

Proof. Let $k_i = t + 1 + n_i$ and $k'_i = t + n'_i$ be the two consecutive trimming times. Since earlier measurements are always trimmed before the later measurements, this guarantees $\Delta k \geq 1$. If $n'_i \geq n_i$, then it takes longer time for the later measurements to be trimmed and thus $\Delta k = 1 + n'_i - n_i$.

Algorithm 3 Trimming CBs using TLs

```
Find the smallest time in the track list: $k_m = \min\left\{k^{i,1}, \ldots, k^{i,N}\right\}$. If all elements with the same time tag in $P^i_k$ are 1's, then
    1. set all these items in the track list to be 0;
    2. remove all corresponding measurements in $i^{\text{th}}$ CB;
    3. increase all time tags equivalent to $k_m$ by 1.
    4. update the track list with items in current CB.
```

Proof. The proof is straightforward by considering Proposition 1, Theorems 1 and 2.

4.3 Complexity of FIFO-DBF

Compared to statistics dissemination, FIFO is generally more communication-efficient for distributed filtering. To be specific, consider a $D \times D$ grid environment with a network of N UGVs, the transmitted data of FIFObetween each pair of UGVs are only the CB of each UGV and the corresponding UGV positions where observations were made, the length of which is $O(N)$. On the contrary, the length of transmitted data for a statistics dissemination approach that transmits unparameterized posterior distributions or likelihood functions is $O(D^2)$, which is in the order of environmental size. Since D is generally much larger than N in applications such as target localization, FIFOrequires much less communication resources.

It is worth noting that, for the static target, each UGV only needs current-step CB to update individual PDFs. Therefore, besides storing its own individual PDF of size $O(M^2)$, only current-step CB of size $O(N)$ is stored in an UGV's memory and all previous CBs can be discarded, which means that the size of needed memory is $O(N + M^2)$. On the contrary, for the moving target, each UGV needs to store a set of measurement history of size $O(N^2)$ and an individual PDF of size $O(M^2)$. Therefore the size of the needed memory for each UGV is $O(M^2 + N^2)$. This is generally larger than that of statistics dissemination-based methods, the memory of which is $O(M^2)$. Besides, additional computation power is needed for LIFO-DBF compared to statistics dissemination-based methods. Therefore, LIFO-DBF sacrifices storage space and computation resource for reducing 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 4. *We can change TLs to track measurements of multiple time steps so that CBs might be trimmed off multiple time steps at each trim. This, however, requires larger communication resource. The maximum size of CBs transmitted by UGVs is $O(NT_m)$, where $T_m = \max_{l,j,i} \{d_{lj} + d_{ji}\}$ and $T_m \leq NT_u$.*

Remark 5. *According to Theorem 2, CBs can grow to undesirable sizes that causes excessive communication burden. An approximate algorithm is to use a time window for 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). An estimator of a state is said to be consistent if it converges in probability to the true value of the state [23]. Consistency is an important metric for stochastic filtering approaches [24] and it differs from the concept of consensus; consensus implies that the estimation results of all sensors converge to a same value, while 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 S is a finite set (e.g. a finely discretized field).

5.1 Static UGVs

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

Theorem 3. *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 LIFO-DBF, i.e.,*

$$\lim_{k\to\infty} P(X_k^{MAP} = x_k^g | \mathbf{z}_{1:k}^i) = 1,\ i = 1, \ldots, N,$$

where

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

Proof. For the purpose of clarity, define the time set of i^{th} UGV, $\mathscr{K}_{j,k}^i$, $j \in \{1, \ldots, N\}$, that contains the time steps of measurement by j^{th} UGV that are contained in B_k^i. According to **??**, it is known that the cardinality of $\mathscr{K}_{j,k}^i$ has following property: $k - (N-1)T_u < |\mathscr{K}_{j,k}^i| \leq k$. Considering the conditional independence of measurements given $x_k^g \in S$, the batch form of DBF at k^{th} step is

$$\begin{aligned} P_{pdf}^i(X_k | B_k^i) &= P_{pdf}^i(X_k | z_{1:k_1^i}^1, \ldots, z_{1:k_N^i}^N) \\ &= \frac{P_{pdf}^i(X_0) \prod_{j=1}^{N} \prod_{l=1}^{k_j^i} P(z_l^j | X_l) P(X_l | X_{l-1})}{\sum_{X_0, \ldots, X_k \in S} P_{pdf}^i(X_0) \prod_{j=1}^{N} \prod_{l=1}^{k_j^i} P(z_l^j | X_l) P(X_l | X_{l-1})}. \end{aligned}$$

Comparing $P_{pdf}^i(x^T | B_k^i)$ with $P_{pdf}^i(x^{T^*} | B_k^i)$ yields

$$\frac{P_{pdf}^i(X_k = x_k | B_k^i)}{P_{pdf}^i(X_k = x_k^g | B_k^i)} = \frac{P_{pdf}^i(x_0) \prod_{j=1}^{N} \prod_{l \in \mathscr{K}_{j,k}^i} P(z_l^j | x_l)}{P_{pdf}^i(x_0^g) \prod_{j=1}^{N} \prod_{l \in \mathscr{K}_{j,k}^i} P(z_l^j | x_l^g)}. \tag{13}$$

Take the logarithm of Eq. (13) and average it over k steps:

$$\frac{1}{k} \ln \frac{P_{pdf}^i(X_k = x_k | B_k^i)}{P_{pdf}^i(X_k = x_k^g | B_k^i)} = \frac{1}{k} \ln \frac{P_{pdf}^i(x_0)}{P_{pdf}^i(x_0^g)} + \sum_{j=1}^{N} \frac{1}{k} \sum_{l \in \mathscr{K}_{j,k}^i} \ln \frac{P(z_l^j | x_k)}{P(z_l^j | x_k^g)}. \tag{14}$$

Since $P_{pdf}^i(x_0)$ and $P_{pdf}^i(x_0^g)$ are bounded, then $\lim_{k\to\infty} \frac{1}{k} \ln \frac{P_{pdf}^i(x_0)}{P_{pdf}^i(x_0^g)} = 0$.

Utilizing the facts: (1) z_l^j are conditionally independent samples from $P(z_l^j | x_l^g)$ and (2) $k - (N-1)T_u < |\mathscr{K}_{j,k}^i| \leq k$, the law of large numbers yields

$$\begin{aligned} \frac{1}{k} \sum_{l=1}^{k_j^i} \ln \frac{P(z_l^j | x_l)}{P(z_l^j | x_l^g)} &\xrightarrow{P} \mathbb{E}_{z_l^j} \left[\frac{P(z_l^j | x_l)}{P(z_l^j | x_l^g)} \right] && (15a) \\ &= \int_{z_l^j} P(z_l^j | x_l^g) \frac{P(z_l^j | x_l)}{P(z_l^j | x_l^g)} dz_l^j && (15b) \\ &= -D_{KL}\left(P(z_l^j | x_l) \| P(z_l^j | x_l^g) \right), && (15c) \end{aligned}$$

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 $\forall P_1, P_2,\ D_{KL}(P_1 \| P_2) \leq 0$, equality holds iff $P_1 = P_2$. This leads to the following conclusion:

$$\begin{aligned} \frac{1}{k} \sum_{l=1}^{k_j^i} \ln \frac{P(z_l^j | x_l)}{P(z_l^j | x_l^g)} &< 0, \quad x_l \neq x_l^g && (16a) \\ \frac{1}{k} \sum_{l=1}^{k_j^i} \ln \frac{P(z_l^j | x_l)}{P(z_l^j | x_l^g)} &= 0, \quad x_l = x_l^g. && (16b) \end{aligned}$$

Then by considering the limiting case of Eq. (14), we can get:

$$\begin{aligned} \lim_{k\to\infty} \frac{1}{k} \ln \frac{P_{pdf}^i(x_l | B_k^i)}{P_{pdf}^i(x_l^g | B_k^i)} &< 0, \quad x_l \neq x_l^g && (17a) \\ \lim_{k\to\infty} \frac{1}{k} \ln \frac{P_{pdf}^i(x_l | B_k^i)}{P_{pdf}^i(x_l^g | B_k^i)} &= 0, \quad x_l = x_l^g. && (17b) \end{aligned}$$

Eq. (17a) and (17b) imply that

$$\frac{P^i_{pdf}(x_l|B^i_k)}{P^i_{pdf}(x^g_l|B^i_k)} \xrightarrow{P} 0, \quad x_l \neq x^g_l \tag{18a}$$

$$\frac{P^i_{pdf}(x_l|B^i_k)}{P^i_{pdf}(x^g_l|B^i_k)} \xrightarrow{P} 1, \quad x_l = x^g_l. \tag{18b}$$

Therefore,

$$\lim_{k\to\infty} P(X^{MAP}_k = x^g_k|B^i_k) = 1.$$

5.2 Moving UGVs

The consistency proof for the case when UGVs are moving is different from the static UGVs case in that each moving UGV makes measurements at multiple positions. We classify UGV measurement positions into two disjoint sets: *infinite-measurement spots* that contain positions where a UGV visits infinitely many times 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 visits infinitely many times as k tends to infinity.

Theorem 4. *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 LIFO-DBF, i.e.,*

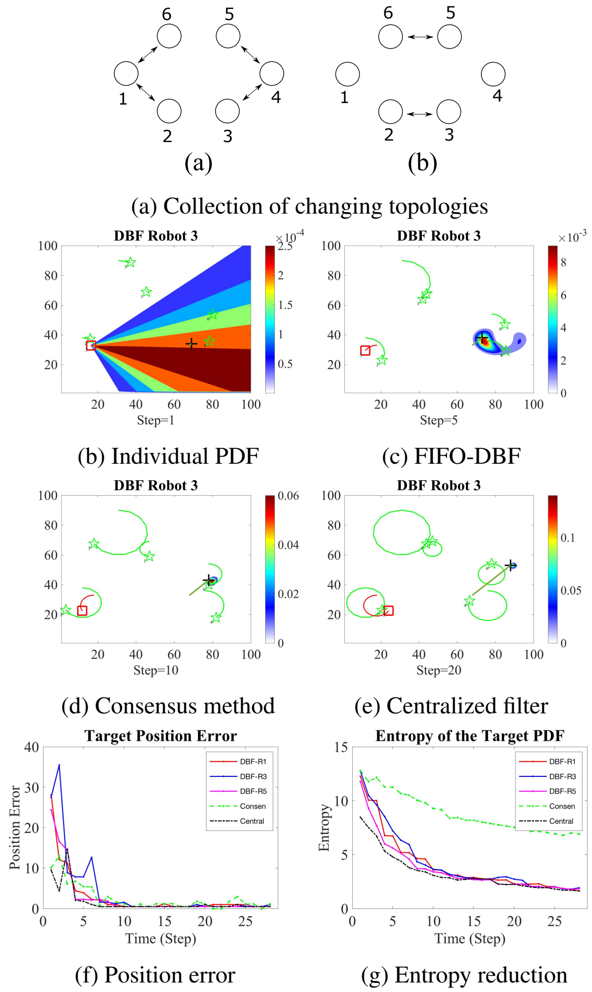
$$\lim_{k\to\infty} P(X^{MAP}_k = x^g_k|B^i_k) = 1,\ i = 1,\dots,N.$$

Proof. The batch form of DBF at k^{th} step is

$$\frac{P^i_{pdf}(X_k = x_k|B^i_k)}{P^i_{pdf}(X_k = x^g_k|B^i_k)} = \frac{P^i_{pdf}(x_0)\prod_{j=1}^{N}\prod_{l\in\mathcal{K}^i_{j,k}} P(z^j_l|x;x^j_l)}{P^i_{pdf}(x^g_0)\prod_{j=1}^{N}\prod_{l\in\mathcal{K}^i_{j,k}} P(z^j_l|x^g;x^j_l)}. \tag{19}$$

The only difference from Eq. (13) is that $P(z^j_l|x;x^j_l)$ in Eq. (19) varies as the UGV moves. For the finite-measurement spots, by referring to Eq. (15), it is easy to know that their contribution to Eq. (14) diminishes when $k\to\infty$. Therefore, proof using Eq. (19) can be reduced to only considering infinite-measurement spots and the rest of proof is similar to that of Theorem 3.

Remark 6. *The assumption of unbiased sensors are important for the consistency of the estimator. In fact, with unknown non-zero bias, the distribution of z^j_l differs from $P(z^j_l|x^g)$, which invalidates the derivation in Eq. (15) and the consistency proof. This assumption also makes intuitive sense. In the extreme case, if each sensor has a very large unknown measurement offset, then the estimated target position of each sensor (without communicating with other sensors) will be very different from each other's. Therefore, no common target position can be correctly obtained when they fuse measurements.*



(a) (b)

(a) Collection of changing topologies

(b) Individual PDF

(c) FIFO-DBF

(d) Consensus method

(e) Centralized filter

(f) Position error

(g) Entropy reduction

Fig. 3: 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.

6 Simulation

An example result is shown in Figure 3. Will generate better simulation later.

7 Conclusion

This paper presents a measurement dissemination-based distributed Bayesian filtering (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 full communication buffers with neighbors and can significantly reduce the transmission burden between each pair of UGVs to scale linearly with the network size. Under the condition that the union of the switching topologies is connected frequently enough, FIFOcan disseminate observations over the network within finite time. By using the track list, the CBs can be trimmed without sacrificing any unused information. The FIFO-based DBF algorithm is then derived to estimate individual probability density function (PDF) for target localization. The consistency of this algorithm is proved by utilizing the law of large numbers, ensuring that each individual estimate of target position converges in probability to the true value. Simulations comparing FIFO-DBF with consensus-based distributed filters (CbDF) and the centralized filter (CF) show that FIFO-DBF achieves similar performance as CF and superior performance over CbDF while requiring less communication resource.

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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

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

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