

Distributed Bayesian Filter for Multi-Robot Network by Using Local-Exchange-of-Observation Strategy

Albert Author¹ and Bernard D. Researcher²

Abstract—This paper proposes a local-exchange-of-observation (LEO) strategy for distributed Bayesian filters (LEO-DBF) in a multi-robot network with the application of distributed estimation of target position. Different from statistics dissemination-based distributed filtering approaches that depends on the exchange of posterior distribution or likelihood function, LEO only requires each robot to transmit observations with local neighboring robots, which reduces the communication burden for the network. With fixed communication topology, LEO guarantees the spread of all robots' observations over the network within finite time, with each robot non-intermittently receiving all other robots' observations via local communication. Two LEO-DBFs are proposed for estimating the position of the static and the moving target, respectively. For the static target, each robot locally fuses the latest knowledge of all robots' observations by only considering the updating step of the Bayesian filter. For the moving target, a triangle matrix of historical observations is maintained by each robot. Upon obtaining the latest available observations of all robots, an iterative Bayesian filtering procedure is applied that alternates between prediction and updating steps. The consistency of LEO-DBF is proved, ensuring the agreement between robots' estimated target position and the actual one. Simulations are generated to demonstrate the effectiveness of LEO-DBF for tracking both static and moving target.

I. INTRODUCTION

Distributed filtering in multi-robot network that focuses on using a group of networked robots to collectively infer the state of an environment has been used for various applications, such as object detection [1], target tracking [2] and environmental monitoring [3]. Several techniques have been developed for distributed filtering. For example, in [4], a distributed Kalman filter (DKF) was proposed for estimating states of linear systems with Gaussian process and measurement noise. Each DKF used low-pass and band-pass consensus filters for the average-consensus of weighted measurements and inverse-covariance matrices. [5] presented a distributed particle filter for target tracking over sensor networks. Gaussian mixture model (GMM) was adopted to approximate the posterior distribution of weighted particles and the parameters of GMM was exchanged via average consensus filter. This study focuses on developing the distributed Bayesian filter (DBF) that is applicable for state estimation of

general nonlinear systems and the proposed DBF is applied to target search and tracking.

The communication topology of multi-robot network plays a vital role in distributed filtering algorithms. Fusion center (FC)-based DBF has been a common structure for distributed filtering, in which local information collected by robots is transmitted (possibly via multi-hopping) to the fusion center for forming global estimation [6], [7]. FC-based DBF is efficient for estimation in that it can collectively utilize all robots' information and thus especially useful for applications that only require information at a single central unit, such as in environmental monitoring.

The other popular structure is the neighborhood(NB)-based DBF, in which no fusion center is used. Instead, each robot only exchanges information with neighboring robots and forms local estimation of the environment state. NB-based DBF is advantageous over FC-based DBF in that no central unit is required, thus suitable for applications in which maintaining communication link between robots and the center is challenging, such as wilderness search and rescue. Besides, estimation can be locally conducted in each robot, which consumes less computation power compared to that in the fusion center. Thus far, works on NB-based DBF have mainly focused on the so-called statistics dissemination-based strategy that each robot actually exchanges posterior distribution or likelihood function to neighboring robots for distributed estimation. For example, Sheng et al.[8] proposed a leader agent-based distributed particle filters with Gaussian Mixer to localize and track multiple moving targets. The distributed particle filters ran on a set of uncorrelated sensor cliques and information among cliques was exchanged for global estimation of targets. In [9], a circular topology that each sensor could only communicate with a fixed neighboring sensor was deployed for parameter estimation of a spatial field. Each sensor generates state estimate based on that of the previous sensor and its own observation and sequentially passes the estimate to its neighbor.

(TODO: may argue that statistics dissemination can be problematic in that true state estimation is not guaranteed.) A popular strategy for statistics dissemination is to use consensus-based approaches that all robots perform distributed filtering simultaneously, exchange statistics with neighbors and executes consensus algorithms, as proposed in [10], [11], [12], for fusion of statistics. For example, Saptarshi et al. [13] presented a Bayesian consensus filter (BCF) that uses logarithmic opinion pool for fusing posterior distributions of the tracked target among neighboring robots. A consensus-based distributed particle filter (DPF)

*This work was not supported by any organization

¹Albert Author is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands albert.author@papercept.net

²Bernard D. Researcher is with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA b.d.researcher@ieee.org

is proposed by Julian et al. [14] for estimating environment state by exchanging and fusing posterior functions of the state among neighbors using a discrete-time linear consensus algorithm.

Despite the popularity of statistics dissemination approaches, exchanging posterior distribution or likelihood functions can consume high communication resources, which may be infeasible for applications in vast area or complex environment, such as marine search, seismological rescue, etc. This study focuses on the strategy of exchanging observation in the neighborhood of each robot for the purpose of achieving a consensus of local target PDFs. Some pioneering studies have been done on measurement dissemination strategies, in which raw or quantized observations are exchanged among robots. For example, Coates et al. [15] used adaptive encoding of observations to minimize communication overhead. Djuric et al. [16] proposed a decentralized particle filter for tracking targets. At each time instant, a subset of robots that are in proximity of the tracked targets share their observations with each other for target position estimation. Another example can be found in [17], in which both observation and statistics are exchanged. In aforementioned works, either communication topology is assumed fully connected that each robot broadcasts observations to all other robots in single transmission step or the communication involves large amount of exchange data.

This paper proposes a local-exchange-of-observation (LEO) strategy for distributed Bayesian filters (DBF) for undirected local topology. Each robot only broadcasts observations to its local neighbors and implements the Bayesian filter based on received observations. The main benefit of LEO is the reduction of communication burden, with the transmission data volume scaling linearly with the robot number, while statistics dissemination-based strategies suffer from the data volume on the order of environmental size. After receiving observations from neighbors, each robot runs Bayesian filter locally for environment state estimation. The proposed LEO-DBF has following properties: (1) For an undirected and connected network with fixed topology, LEO guarantees the spread of all robots' observations among the network via multi-hopping, with each robot non-intermittently receiving (delayed) observations of all other robots. (2) LEO-DBF can ensure consistency of the state estimate, thus guaranteeing consensus. The consistency property refers to the agreement between the robot's state estimate and the true state. In this study, a formal proof of the consistency of LEO-DBF is provided.

The rest of this paper is organized as follows: The distributed target position estimation is formulated in Section II. The LEO-DBF algorithm is described in Section III followed by the proof of its consistency and consensus in Section IV. Simulation results of LEO-DBF and conclusions are presented in Section V and Section VI, respectively.

II. PROBLEM FORMULATION

Consider a network of N robots in bounded two-dimensional space S . Each robot is equipped with a binary

sensor for environmental perception. Due to the limit of communication range, each robot can only exchange observations with its neighbors. Bayesian filter is run locally on each robot for environment state estimation based on its own and received observations.

A. Probabilistic Model of Binary Sensor

The binary sensor only gives two types of observation: 1 if the target is detected, and 0 if no target is detected. The observation of i^{th} sensor at k^{th} time step is denoted as z_k^i . The likelihood function that the target is detected is:

$$P(z_k^i = 1 | x_k^T) \in [0, 1], \quad x_k^T \in X^T \subseteq S \quad (1)$$

where x_k^T is the target position, X^T is the set of all possible target positions.

Correspondingly, the likelihood function that no target is detected is:

$$P(z_k^i = 0 | x_k^T) = 1 - P(z_k^i = 1 | x_k^T) \quad (2)$$

The combination of (1) and (2) is actually binary sensor model, parameterized by x_k^T . The goal of Bayesian filter is to estimate the actual target position x_k^{T*} , with which all observations are generated.

The commonly used likelihood functions for binary sensor are Gaussian function [18], [19] and step function [20]. In addition, LEO also works for other types of sensors, including imaging sensors, such as laser scanners [21] and cameras [22].

Remark 1: The current observation of each robot is conditionally independent from both its own past observations and those of other robots, given the information of target position.

Remark 2: This study is applicable for both homogenous and heterogeneous binary sensors. A homogeneous model can simplify the analysis of completeness, while the heterogeneous model is more close to real sensing characteristics.

B. Graphical model of communication topology

Consider an undirected and connected graph $G = (V, E)$, where $V = \{1, \dots, N\}$ is the set of robots and $E = V \times V$ denotes the edge set. The adjacency matrix $M = (m_{i,j})$ describes the communication topology of G :

$$m_{i,j} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{if } (i, j) \notin E \end{cases}, \quad (3)$$

where $m_{i,j}$ denotes the entity of adjacency matrix. The notation $m_{i,j} = 1$ indicates that a communication link exists between i^{th} and j^{th} robot and $m_{i,j} = 0$ indicates no communication.

The *direct neighborhood* of i^{th} robot is defined as $\mathcal{N}_i = \{j | m_{i,j} = 1, \forall j \in \{1, \dots, N\}\}$. All the robots in \mathcal{N}_i can directly exchange information with i^{th} robot. In addition to *direct neighborhood*, another set, called *available neighborhood*, is defined as \mathcal{Q}_i , which contains indices of robots whose observation is received by the i^{th} robot given a specific observation exchange algorithm. Note that in general $\mathcal{N}_i \subseteq \mathcal{Q}_i$, but if only single-hopping is allowed, $\mathcal{Q}_i = \mathcal{N}_i$. Fig. 1

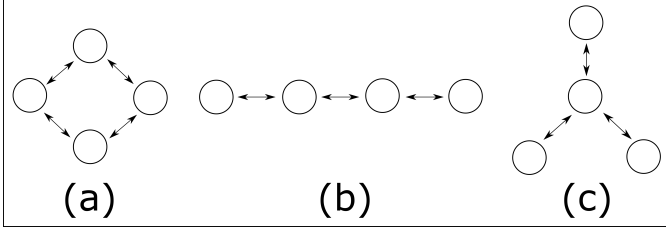


Fig. 1: Three types of topologies: (a)circular topology; (b)serial topology; (c)star topology

illustrates three types of typical topologies: circular [9], serial [23], and star [8]. All of them are undirected and connected topologies.

C. Distributed Bayesian Filter for Multiple Robots

The generic Distributed Bayesian Filter (DBF) is introduced in this section, which was also stated in [24] and [25]. Each robot has its individual estimation of the probability density function (PDF) of target position, called *individual PDF*. **(TODO: should the individual PDF always be italic throughout the paper?)** The i^{th} individual PDF at time k is defined as $P_{PDF}^i(x_k^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i})$. Before starting DBF, the individual PDF is initialized by the prior function $P_{PDF}^i(x_0^T | z_0^i, z_0^{\mathcal{Q}_i}) = P(x_0^T)$, given all available prior information including past experience and environmental knowledge. Then the individual PDF is recursively estimated by two steps, i.e., prediction step and updating step, based on the observations of the robots in \mathcal{Q}_i .

1) *Prediction*: The i^{th} individual PDF at time $k-1$ is known, denoted as $P_{PDF}^i(x_{k-1}^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i})$. At time k , the prior individual PDF is first predicted forward by using the Chapman-Kolmogorov equation:

$$P_{PDF}^i(x_k^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i}) \quad (4)$$

$$= \int P(x_k^T | x_{k-1}^T) P_{PDF}^i(x_{k-1}^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i}) dx_{k-1}^T \quad (5)$$

where $P(x_k^T | x_{k-1}^T)$ is a Markov motion model of the target, independent of robot states. This model describes the state transition probability of the target from a prior state x_{k-1}^T to posterior state x_k^T . Note that the target can be static in many search applications, such as the indoor search for stationary objects[26]. For a static target, its Markov motion model is simplified to be

$$P(x_k^T | x_{k-1}^T) = \begin{cases} 1 & \text{if } x_k^T = x_{k-1}^T \\ 0 & \text{if } x_k^T \neq x_{k-1}^T \end{cases} \quad (6)$$

and (4) can be reduced to $P_{PDF}^i(x_k^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i}) = P_{PDF}^i(x_{k-1}^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i})$.

2) *Updating*: At time step k , the i^{th} individual PDF is then updated by Bayes' formula using the latest available

observations at time k :

$$P_{PDF}^i(x_k^T | z_{1:k}^i, z_{1:k}^{\mathcal{Q}_i}) \quad (7)$$

$$= K_i P_{PDF}^i(x_k^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i}) P(z_k^i | x_k^T) \prod_{j \in \mathcal{Q}_i} P(z_k^j | x_k^T) \quad (8)$$

where K_i is a normalization factor, given by:

$$K_i = 1 / \int P_{PDF}^i(x_k^T | z_{1:k-1}^i, z_{1:k-1}^{\mathcal{Q}_i}) P(z_k^i | x_k^T) \prod_{j \in \mathcal{Q}_i} P(z_k^j | x_k^T) dx_k^T \quad (9)$$

where $P_{PDF}^i(x_k^T | z_{1:k}^i, z_{1:k}^{\mathcal{Q}_i})$ is called posterior individual PDF; $P(z_k^i | x_k^T)$ is the likelihood function of observation z_k^i , described in (1) and (2).

III. DISTRIBUTED BAYESIAN FILTER VIA LOCAL EXCHANGE OF OBSERVATION

This study proposes a Distributed Bayesian Filtering (DBF) algorithm based on LEO strategy, shorted as LEO-DBF. The LEO only uses the local communication within the neighborhood of each robot, but allows to broadcast observations of each robot to any other nodes by multi-hopping along the shortest path in the undirected and connected network. The theoretical analysis show that LEO-DBF can ensure the consistency and consensus of individual PDF while requiring much less communication burden than any statistics dissemination-based DBFs.

A. Algorithm for Local Exchange of observations (LEO)

Under the LEO, each robot contains a communication buffer (CB) to store its latest knowledge of the observations of all robots:

$$\mathbf{z}_k^i = [z_{k_1}^1, \dots, z_{k_N}^N] \quad (10)$$

where $z_{k_j}^j$ represents the observation made by j^{th} robot at time k_j^i . Note that under LEO, $\mathcal{Q}_i = \{1, \dots, N\}$. At time k , $z_{k_j}^j$ is received and stored in i^{th} robot CB, in which k_j^i is the latest available observation time of j^{th} robot known by i^{th} robot. Due to the communication delay of multi-hopping, $k_j^i < k$ always holds in practice.

The **LEO algorithm** is stated as follows:

(1) Initialization: The buffer of i^{th} robot is initialized when $k = 0$:

$$z_{k_j}^j = \emptyset, k_j^i = 0, j = 1 : N \quad (11)$$

(2) Repeat the following steps for i-th robot until stop:

(2.1) Receiving Step:

The i^{th} robot receives all CBs of its neighboring robots. The received CBs are totally $|\mathcal{N}_i|$ groups, each of which corresponding to the $(k-1)$ -step CB of a robot in \mathcal{N}_i . The received CB from l^{th} ($l \in \mathcal{N}_i$) robot is denoted as

$$\mathbf{z}_{k-1}^l = [z_{k-1_1}^1, \dots, z_{k-1_N}^N], l \in \mathcal{N}_i \quad (12)$$

(2.2) Observation Step:

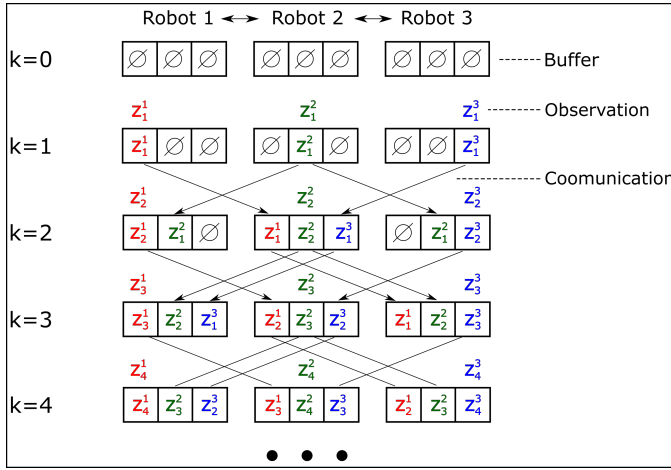


Fig. 2: Example of LEO with three robots using linear communication topology

The i^{th} robot updates $z_{k^i}^j$ by its own observation at current step:

$$z_{k^i}^j = z_k^i, \quad k_j^i = k, \quad \text{if } j = i. \quad (13)$$

(2.3) Comparison Step:

The i^{th} robot updates other elements of its own CB, i.e., $z_{k^i}^j$, $j \neq i$, by selecting the latest information among all received CBs from \mathcal{N}_i . For all $j \neq i$,

$$l_{\text{latest}} = \operatorname{argmax}_{l \in \mathcal{N}_{i,i}} \left\{ (k-1)_j^i, (k-1)_j^l \right\}$$

$$z_{k^i}^j = z_{(k-1)_j^l}^j, \quad k_j^i = (k-1)_j^l$$

(2.4) Sending Step: The i^{th} robot broadcasts its updated CB to all of its neighbors defined in \mathcal{N}_i .

Fig. 2 illustrates the LEO algorithm with 3 robots using a serial topology. Two facts can be observed in Figure 2: (1) all robot CBs are filled within 3 steps, which means under LEO each robot has a maximum delay of 2 steps when receiving observations from other robots; (2) after filled, the updating of CBs are non-intermittent, which means each robot continuously receives newer observations of other robots. Extending the two facts to a network of N robots, we have the following proposition:

Proposition 1: For an undirected and connected network of N robots with fixed topology, LEO uses the shortest path(s) between i and j to exchange observation, i.e., the delay $\tau_{i,j}$ between i and j is equivalent to the length of their shortest path(s).

Proof: Without loss of generality, assume that there is a unique shortest path between i and j , denoted by $T_{n^*}^{j,i} = (v_1, \dots, v_{n^*})$, with $v_1 = j$, $v_{n^*} = i$, $v_{m+1} \in \mathcal{N}_{v_m}$. Then, the distance between i and j is $d(j,i) = n^* - 1$. The following mathematical induction will prove ?? 1.

Step (1): For $d(j,i) = 1$, this means $j \in \mathcal{N}_i$ and j can directly send z_k^j to i . Then z_k^j is stored in i^{th} CB at time $k+1$, i.e. $\tau_{i,j} = 1$. ?? 1 holds for $d(j,i) = 1$, $i, j \in \{1, \dots, N\}$.

Step (2): Suppose that ?? 1 holds for $d(j,i) = s$, $s \geq 2$. Then for $d(j,i) = s+1$, i.e., $n^* = s+2$, by the Bellman's principle of optimality, the path $T_{n^*-1}^{j,i} = (v_1, \dots, v_{n^*-1})$ is the shortest path between j and i , where $v_{n^*-1} = l$ and $i \in \mathcal{N}_l$. The assumption that ?? 1 holds for $d(j,i) = s$ implies that z_k^j is received and stored in l^{th} robot's CB at time $k+s$. Since $i \in \mathcal{N}_l$, i^{th} robot receives z_k^j at $k+s+1$. Therefore $\tau_{i,j} = s+1$. For any other path $T_n^{j,i} = (v_1, \dots, v_n)$ with $n > n^*$, z_k^j cannot be received by i^{th} robot at $k+s+1$. This proves the ?? 1 for $d(j,i) = s+1$. ■

Corollary 1: For the same topology assumption in ?? 1, all elements in z_k^i under LEO become filled when $k \geq N$.

Proof: In a network of N robots, the maximal length of shortest paths is no greater than $N-1$. Based on ?? 1, $\tau_{i,j} \leq N-1$ and thus all elements of z_k^i become filled when $k \geq N$. ■

Corollary 2: For the same topology in ?? 1, once all elements in z_k^i are filled, the updating of each element is non-intermittent.

Proof: For a network with fixed topology, the shortest path between any nodes is fixed. Based on ?? 1, $\tau_{i,j}$ is constant, and then the updating of each element in z_k^i is non-intermittent. ■

Remark 3: Compared to statistics dissemination, LEO is a more communication-efficient approach for distributed filtering. To be specific, consider an $M \times M$ grid environment with a network of N robots, the transmitted data of LEO between each pair of robots are only the CB of each robot, the length of which is $O(N)$. On the contrary, the length of transmitted data for a statistics-dissemination approach is $O(M^2)$, which is the size of environment grid. Since M is generally much larger than N in target search and tracking community, LEO requires much less communication resources than the statistics-dissemination approaches.

B. Algorithm of LEO-DBF for a Static Target

This section gives the LEO-DBF for a static target. Each robot stores last-step individual PDF, i.e., $(k-1)^{\text{th}}$ step. The assumption of static target can simplify the Bayesian filter, in which the prediction step is unnecessary. Therefore, the i^{th} individual PDF is only updated by

$$P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i) = K_i P_{PDF}^i(x^T | \mathbf{z}_{1:k-1}^i) \prod_{j=1}^N P(z_{k^i}^j | x^T) \quad (14)$$

where

$$K_i = 1 / \int P_{PDF}^i(x^T | \mathbf{z}_{1:k-1}^i) \prod_{j=1}^N P(z_{k^i}^j | x^T) dx^T$$

C. Algorithm of LEO-DBF for a Moving target

This section gives the LEO-DBF for a moving target. Instead of storing last-step PDF, each robot maintains an individual PDF for the time step $(k-N)$ and a triangular matrix of history observations from $(k-N+1)^{\text{th}}$ to k^{th} steps. The i^{th} individual PDF is then alternatively predicted and updated by using aforementioned Bayesian filter ((4)

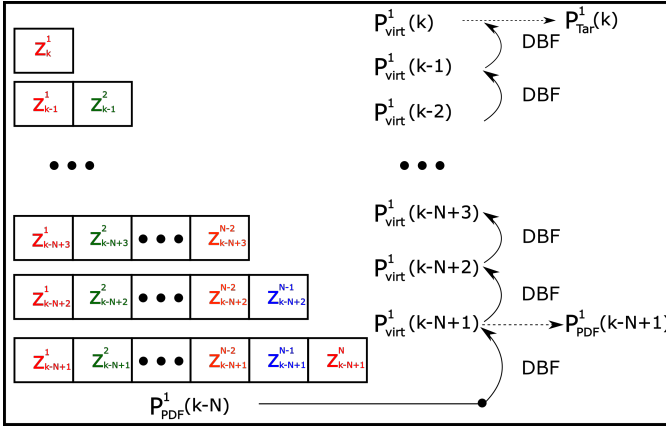


Fig. 3: Example of LEO-DBF for 1st robot at time k . The current individual PDF is $P_{PDF}^1(x_{k-N}^T | z_{1:k-N}^1, \dots, z_{1:k-N}^N)$, denoted as $P_{PDF}^1(k-N)$ in the figure. The robot first calculates $P_{virt}^1(k-N+1)$ using DBF and stores it as $P_{PDF}^1(k-N+1)$. Repeating DBF until obtaining $P_{PDF}^1(k)$, which is then used as the target PDF estimation of i -th robot

and (7)) from $(k-N)^{\text{th}}$ to k^{th} step. Fig. 3 illustrates the LEO-DBF procedure for the 1st robot.

(TODO: shorten the formula length) For the i^{th} robot at k^{th} step, the LEO-DBF for moving target is:

- 1) **The stored individual PDF for $(k-N)^{\text{th}}$ step is:**
 $P_{PDF}^i(x_{k-N}^T | z_{1:k-N}^1, \dots, z_{1:k-N}^N)$.
- 2) **Initialize a virtual PDF by assigning the individual PDF to it:**
 $P_{virt}^i(x_{k-N}^T) = P_{PDF}^i(x_{k-N}^T | z_{1:k-N}^1, \dots, z_{1:k-N}^N)$
- 3) **From $\xi = 1$ to N , repeat two steps of Bayesian filtering:**

a) Prediction

$$P_{virt}^{pre}(x_{k-N+\xi}^T) = \int P(x_{k-N+\xi}^T | x_{k-N+\xi-1}^T) P_{virt}(x_{k-N+\xi-1}^T) dx_{k-N+\xi-1}^T$$

b) Updating

$$P_{virt}(x_{k-N+\xi}^T) = K_{\xi} P_{virt}^{pre}(x_{k-N+\xi}^T) \prod_{j \in \Omega_{k-N+\xi}^i} P(z_{k-N+\xi}^j | x_{k-N+\xi}^T)$$

$$K_{\xi} = 1 / \int P_{virt}^{pre}(x_{k-N+\xi}^T) \prod_{j \in \Omega_{k-N+\xi}^i} P(z_{k-N+\xi}^j | x_{k-N+\xi}^T) dx_{k-N+\xi}^T \rightarrow \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \end{cases}, k \rightarrow \infty \quad (16)$$

- 4) **Store the first-step virtual PDF as the $(k-N+1)^{\text{th}}$ individual PDF:**

$$P_{PDF}^i(x_{k-N+1}^T | z_{1:k-N+1}^1, \dots, z_{1:k-N+1}^N) = P_{virt}(x_{k-N+1}^T)$$

where $\Omega_{k-N+\xi}^i$ denotes the index set of robots whose observation at time $k-N+\xi$ is stored in i^{th} robot's CB. The target PDF estimation of i^{th} robot at current step is $P_{virt}(x_k^T)$.

Remark 4: For the static target, each robot only needs current step CB to update individual PDFs. Therefore, except storing individual PDFs, all historical CBs can be discarded and only current-step CB is stored in robot memory, the

length of which is $O(N)$. On the contrary, for the moving target, each robot needs to store a triangular matrix of history observation (except current step CB) with $N \times N$ dimension and an $M \times M$ individual PDF, which means that the length of occupied memory in each robot is $O(M^2 + N^2)$.

IV. PROOF OF CONSISTENCY AND CONSENSUS

This section presents a consistency and consensus proof of LEO-DBF. Only the scenarios for the static target are presented, including both static robots and moving robots. The proof for LEO-DBF for moving target is similar to that of static target by assuming that the dynamic model of the target is accurately known, but with more complicated algebraic manipulation.

Assume that X^T is finite. Define an *equi-parameter* set $X_{eq}^T \subseteq X^T$ such that

$$P(z_k | x_1^T) = P(z_k | x_2^T), \forall x_1^T, x_2^T \in X_{eq}^T, \forall z_k \in \{0, 1\}, \quad (15)$$

i.e., the probability for generating the same observation z_k is equivalent among sensor models with different parameters in X_{eq}^T . Since X^T is finite, X_{eq}^T is also finite. Let $X_{eq,1}^T, \dots, X_{eq,u}^T$ denote all *equi-parameter* sets that partition X^T such that following properties hold:

- 1) $\bigcup_{i=1}^u X_{eq,i}^T = X^T$
- 2) $X_{eq,i}^T \cap X_{eq,j}^T = \emptyset, i, j \in \{1, \dots, u\}, i \neq j$.

Without loss of generality, assume $x^{T*} \in X_{eq,u}^T$, where x^{T*} denotes the actual location of the target.

(TODO: double check if the following examples for equi-parameter set are accurate)

Remark 5: the *equi-parameter* set depends on the property of the sensor. For example, for a laser scanner with high-fidelity sensing capability, each *equi-parameter* set contains only a small number of target positions.

A. Proof for static robots

The consistency of LEO-DBF for static robots is stated as follows:

Theorem 1: For static robots, each individual PDF by LEO-DBF converges to $X_{eq,1}^T$ when the number of observations goes to infinity, i.e.

$$P_{PDF}^i(x^T | z_{1:k}^1, \dots, z_{1:k}^N) \rightarrow \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \end{cases}, k \rightarrow \infty \quad (16)$$

where $z_{1:k}^i = (z_{1:k_1}^1, \dots, z_{1:k_N}^N)$, $\{k_1, \dots, k_N\}$ are the timestamps of the i^{th} robot's latest knowledge of all robots' observations.

Proof: Considering the conditional independence of observations z_k^j for given $x^T \in X_{eq,i}^T$, the batch form of DBF at k^{th} step is:

$$P_{PDF}^i(x^T | z_{1:k}^1) = P_{PDF}^i(x^T | z_{1:k_1}^1, \dots, z_{1:k_N}^N) \quad (17a)$$

$$= \frac{P_{PDF}^i(x^T) \prod_{j=1}^N \prod_{l=1}^{k_j} P(z_l^j | x^T)}{\sum_{x^T \in X^T} \prod_{j=1}^N \prod_{l=1}^{k_j} P(z_l^j | x^T)} \quad (17b)$$

where P_{PDF}^i is initial i^{th} individual PDF. It is known from ?? 1 and ?? 2 that $k - N < k_j \leq k$.

Compare $P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)$ with $P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)$:

$$\frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = \frac{P_{PDF}^i(x^T) \prod_{j=1}^N \prod_{l=1}^{k_j} P(z_l^j | x^T)}{P_{PDF}^i(x^{T*}) \prod_{j=1}^N \prod_{l=1}^{k_j} P(z_l^j | x^{T*})} \quad (18)$$

Take the logarithm of (18) and average it over k steps:

$$\frac{1}{k} \ln \frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = \frac{1}{k} \ln \frac{P_{PDF}^i(x^T)}{P_{PDF}^i(x^{T*})} + \sum_{j=1}^N \frac{1}{k} \sum_{l=1}^{k_j} \ln \frac{P(z_l^j | x^T)}{P(z_l^j | x^{T*})} \rightarrow \infty. \quad (19)$$

Since $P_{PDF}^i(x^T)$ and $P_{PDF}^i(x^{T*})$ are bounded,

$$\lim_{k \rightarrow \infty} \frac{1}{k} \ln \frac{P_{PDF}^i(x^T)}{P_{PDF}^i(x^{T*})} \rightarrow 0 \quad (20)$$

The binary observations subject to Bernoulli distribution $B(1, p_j)$, yielding

$$P_{PDF}^i(z_k^j | x^T) = p_j^{z_k^j} (1 - p_j)^{1 - z_k^j} \quad (21)$$

where $p_j = P(z_k^j = 1 | x^T)$. Utilizing the fact that (1) z_l^j are conditionally independent samples from $B(1, p_j^*)$, and (2) $k - N < k_j \leq k$, the law of large numbers yields

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{l=1}^{k_j} z_l^j = p_j^*, \quad \lim_{k \rightarrow \infty} \frac{1}{k} (k_j - \sum_{l=1}^{k_j} z_l^j) = 1 - p_j^* \quad (22)$$

where $p_j^* = P(z_k^j = 1 | x^{T*})$. Then,

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{l=1}^{k_j} \ln \frac{P(z_l^j | x^T)}{P(z_l^j | x^{T*})} = p_j^* \ln \frac{p_j}{p_j^*} + (1 - p_j^*) \ln \frac{1 - p_j}{1 - p_j^*} \quad (23)$$

Note that the right-hand side of (23) achieves maximum if and only if $p_j = p_j^*$. Considering (20) and (23), the limit of (19) can be obtained:

$$\lim_{k \rightarrow \infty} \frac{1}{k} \ln \frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = \sum_{j=1}^N p_j^* \ln \frac{p_j}{p_j^*} + (1 - p_j^*) \ln \frac{1 - p_j}{1 - p_j^*} \quad (24)$$

It is known from (24) that

1) When $p_j \neq p_j^*$, that is $x^T \notin X_{eq,1}^T$,

$$\lim_{k \rightarrow \infty} \frac{1}{k} \ln \frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} < 0, \text{ thus}$$

$$\lim_{k \rightarrow \infty} \ln \frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = 0$$

2) When $p_j = p_j^*$, that is $x^T \in X_{eq,1}^T$,

$$\lim_{k \rightarrow \infty} \frac{1}{k} \ln \frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = 0, \text{ thus}$$

$$\lim_{k \rightarrow \infty} \ln \frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = 1$$

B. Proof for moving sensors

The difficulty for moving robots lies on ****. The main idea is to classify the states etc. . Before stating main theorem, the following lemma is introduced:

Lemma 1: For a set of robots with a collection of finite possible states Y^R , there exists at least one state for each robot that it visits for infinite times as k tends to infinity.

Proof: Let $n_j^{i,k}$ denote the times that i^{th} robot visits j^{th} state in Y^R up to time k . Then, $\sum_j n_j^{i,k} = k$. It is

straightforward to see that $\exists n_j^{i,k}$, such that $n_j^{i,k} \rightarrow \infty$, as $k \rightarrow \infty$. The consistency of LEO-DBF for moving robots is stated as follows:

Theorem 2: Under the condition of moving robots, each individual PDF by LEO-DBF converges to $X_{eq,1}^T$ when the number of observations goes to infinity, i.e.,

$$P_{PDF}^i(x^T \in X_{eq,i}^T | \mathbf{z}_{1:k}^i) \rightarrow \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \end{cases}, \quad k \rightarrow \infty \quad (25)$$

where $\mathbf{z}_{1:k}^i = (z_{1:k_1}^1, \dots, z_{1:k_N}^N)$, $\{k_1, \dots, k_N\}$ are the timestamps of i^{th} robot's latest knowledge of all robots' observations.

Proof: Let $M \subseteq \{1, \dots, k_j\} \times Y^R$ denote the set of (time, position) pairs to indicate j^{th} robot's state at the corresponding time of observation.

Similar to ?? 1, the batch form of DBF at k^{th} step is:

$$\frac{P_{PDF}^i(x^T | \mathbf{z}_{1:k}^i)}{P_{PDF}^i(x^{T*} | \mathbf{z}_{1:k}^i)} = \frac{P_{PDF}^i(x^T) \prod_{j=1}^N \prod_{l \in M^i} P(z_l^j | x^T)}{P_{PDF}^i(x^{T*}) \prod_{j=1}^N \prod_{l \in M^i} P(z_l^j | x^{T*})} \quad (26)$$

The only difference is that $P(z_l^j | x^T)$ needs to be grouped according to the robot state. For each robot, there exists at least one state from which the robot obtains infinite observations as $k \rightarrow \infty$, according to ?? 1. We can classify all the states into finite-observation states and infinite-observation states. For the former, it is straightforward to know that their contribution to (24) is zero. Therefore, the proof of (26) can be reduced to infinite-observation states, which is similar to ?? 1. ■

Remark 6: Since $X_{eq,1}^T$ is unique, the consistency of LEO-DBF means the consensus of individual PDFs.

Remark 7: The consistency property differentiates LEO-DBF from the statistics dissemination-based filtering approaches. In fact, the statistics dissemination-based methods can ensure the convergence of the state estimate among robots. However, there's no guarantee whether the agreed estimate is close to the true target position.

V. SIMULATION

This section simulates three searching scenarios in order to demonstrate the effectiveness of LEO-BDF. In all scenarios, six robots are utilized and each robot is equipped with

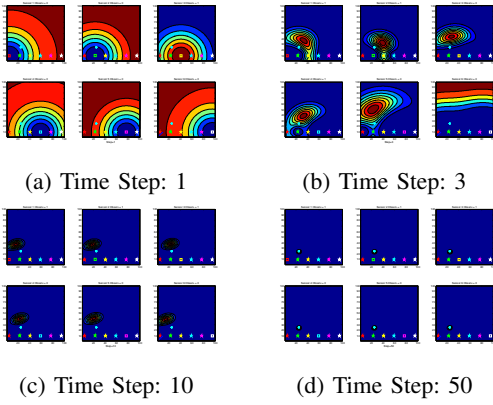


Fig. 4: Static robots' individual PDFs of the static target.

a binary sensor. All sensors are modeled with identical Gaussian functions:

$$P(z = 1|x^T; x^s) = \exp \left\{ -\frac{1}{2}(x^T - x^s)^T \Sigma^{-1}(x^T - x^s) \right\} \quad (27)$$

$$P(z = 0|x^T; x^s) = 1 - P(z = 1|x^T; x^s) \quad (28)$$

where x^s denotes the robot position where current observation is obtained.

The first scenario consists of six static robots and single static target, which acts as a proof of concept of the LEO strategy for static target case. The second scenario subsequently deals with six moving robots for searching the single static target. Robot positions are randomly generated at each time step. The third scenario is presented that contains six moving sensors and one moving target. A single-integrator dynamics is used for the target motion model.

A. Static Robots, Static Target

The positions of six static sensors are shown as stars in (TODO: use cite: Figure 5). Each sensor constantly receives binary observations of the target. LEO-DBF (14) is implemented on each robot for target position estimation.

Fig. 4 shows the estimation results of the static target. After the initial observation, each sensor forms a circular individual PDF, centered at the corresponding sensor position. As more observations are received, the posterior individual PDF concentrates on the true location of the target. Figure 6 shows the decrease of the entropy of the posterior distribution (TODO: add a ref), indicating the uncertainty reduction of the estimated target position.

B. Moving Robots, Static Target

The six robots move within the field to estimate the target position. The motion planning of robots for effective target search has received much attention in the past years. In this work, the robot positions are randomly generated at each time in order to demonstrate the effectiveness of the LEO-DBF approach. Readers interested in robot motion planning can refer to [22], [25].

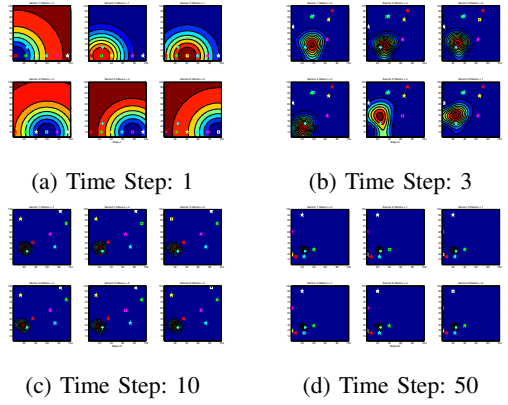


Fig. 5: Moving robots' individual PDFs of the static target.

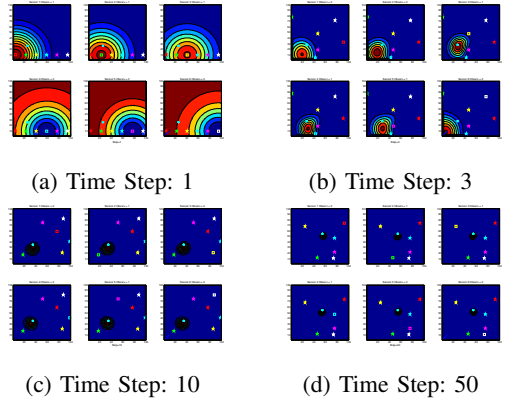


Fig. 6: Moving robots' individual PDFs of the moving target.

Fig. 5 shows the estimation results over time. Similar to the results in Section V-A, the posterior individual PDF concentrates to the true target location. Figure 8 shows the decrease of the entropy of the posterior distribution.

C. Moving Robots, Moving Target

The target in this scenario moves on the horizontal plane and the dynamics is given by a single-integrator model:

$$x^T(k+1) = x^T(k) + v \quad (29)$$

where v is the constant velocity of the target.

The LEO-DBF described in Section III-C is used as the distributed Bayesian filters. Fig. 6 shows the estimation results of the moving target. It can be noticed that similar to the case of static target, the posterior individual PDFs concentrate to the true target location at each time, even when the target constantly moves.

VI. CONCLUSION

In this study, we proposed the local-exchange-of-observation (LEO) strategy for distributed Bayesian filters (LEO-DBF) in a multi-robot network with the application of distributed estimation of target position in the environment. With fixed communication topology, LEO guarantees the spread of all robots' observations within the network. Once elements in communication buffer (CB) gets filled, each

robot can received and update its CB non-intermittently under LEO. Two LEO-DBFs are proposed for estimating the position of the static and the moving target, respectively. For the static target, each robot locally fuses the latest knowledge of all robots' observations by only considering the updating step of the Bayesian filter. For the moving target, a triangle matrix of historical observations is maintained by each robot. Upon obtaining the latest available observations of all robots, an iterative Bayesian filtering procedure is applied that alternates between prediction and updating steps. The consistency of LEO-DBF is proved by showing the asymptotic concentration of posterior individual PDF to the *equi-parameter set* containing the actual state, ensuring the agreement between robots' state estimate using LEO-DBF and the actual environment state. Simulations demonstrated the effectiveness of LEO-DBF for tracking both static and moving target.

Future work may include several extensions to the proposed LEO-DBF. LEO under switching topology can leads to unpredictable delay and intermittent transmission. Therefore consensus analysis of individual PDFs requires further analysis. In addition, combining LEO-DBF with robot motion planning is promising for more effective estimation of target state in the environment.

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] J. Beaudeau, M. F. Bugallo, and P. M. Djurić, "Target tracking with asynchronous measurements by a network of distributed mobile agents," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3857–3860.
- [3] X. Cao, J. Chen, Y. Zhang, and Y. Sun, "Development of an integrated wireless sensor network micro-environmental monitoring system," *ISA transactions*, vol. 47, no. 3, pp. 247–255, 2008.
- [4] 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*. IEEE, 2005, pp. 8179–8184.
- [5] D. Gu, "Distributed particle filter for target tracking," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3856–3861.
- [6] 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*. IEEE, 2006, pp. 1–4.
- [7] 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.
- [8] 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*. IEEE Press, 2005, p. 24.
- [9] S. S. Ram, A. Nedic, and V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*. IEEE, 2007, pp. 582–586.
- [10] 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.
- [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] A. Jadbabaie, J. Lin, *et al.*, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, 2003.
- [13] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using bayesian consensus filtering," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 634–641.
- [14] 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.
- [15] 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.
- [16] P. M. Djurić, J. Beaudeau, and M. F. Bugallo, "Non-centralized target tracking with mobile agents," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5928–5931.
- [17] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized sensor fusion with distributed particle filters," in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 493–500.
- [18] D. Bonnie, S. Candido, T. Bretl, and S. Hutchinson, "Modelling search with a binary sensor utilizing self-conjugacy of the exponential family," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3975–3982.
- [19] C. Liu, S.-Y. Liu, E. L. Carano, and J. K. Hedrick, "A framework for autonomous vehicles with goal inference and task allocation capabilities to support peer collaboration with human agents," in *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2014, pp. V002T30A005–V002T30A005.
- [20] P. M. Djurić, M. Vemula, and M. F. Bugallo, "Target tracking by particle filtering in binary sensor networks," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2229–2238, 2008.
- [21] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. IEEE, 2003, pp. 206–211.
- [22] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *Robotics & Automation Magazine, IEEE*, vol. 16, no. 2, pp. 35–42, 2009.
- [23] I. Bahceci, G. Al-Regib, and Y. Altunbasak, "Serial distributed detection for wireless sensor networks," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*. IEEE, 2005, pp. 830–834.
- [24] O. Hlinka, F. Hlawatsch, and P. Djurić, "Distributed sequential estimation in asynchronous wireless sensor networks," 2015.
- [25] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive bayesian search-and-tracking using coordinated uavs for lost targets," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2521–2526.
- [26] M. Kulich, L. Preucil, and J. J. Miranda Bront, "Single robot search for a stationary object in an unknown environment," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5830–5835.