Connectivity-Preserving Distributed Informative Path Planning for Mobile Robot Networks

Binh Nguyen, Truong X. Nghiem, Linh Nguyen, Hung M. La, and Thang Nguyen, *Member, IEEE*

***Abstract*—This letter addresses the distributed informative path planning (IPP) problem for a mobile robot network to optimally explore a spatial field. Each robot is able to gather noisy environmental measurements while navigating the environment and build its own model of a spatial phenomenon using the Gaussian process and local data. The IPP optimization problem is formulated in an informative way through a multi-step prediction scheme constrained by connectivity preservation and collision avoidance. The shared hyperparameters of the local Gaussian process models are also arranged to be optimally computed in the path planning optimization problem. By the use of the proximal alternating direction method of multiplier, the optimization problem can be effectively solved in a distributed manner. It theoretically proves that the connectivity in the network is maintained over time whilst the solution of the optimization problem converges to a stationary point. The effectiveness of the proposed approach is verified in synthetic experiments by utilizing a real-world dataset.**

***Index Terms*—Path planning for multiple mobile robots or agents, integrated planning and learning, distributed robot systems, distributed learning, informative path planning.**

I. Introduction

OVER the past decades, mobile robot networks have been widely used to perform tasks in many applications, including exploring oceanic and terrestrial ecosystems, tracking wildfires and toxic pollutants, and assessing conditions of critical infrastructures [1], [2], [3]. By taking advantage of mobility and wireless communication, they are practically useful for exploring an unknown environment, especially for monitoring a spatial phenomenon. More importantly, given the environmental measurements the robots have gathered along their way, if a model of the spatial field is learned, the network can predict the field at any unobserved locations in the environment.

However, the fundamental challenge is how to drive the mobile robots to the most informative positions in the environment so that the additional information gained by the spatial field measurements in the network is maximal. This challenge is often known as an *informative path planning* (IPP) problem. In practice, this problem is usually coupled with network constraints such as battery limitation, wireless communication range, and obstacle avoidance. In a large-scale network, it is highly expected that each robot should have its own informative path planner, which allows it to decide where to move in the next steps in a distributed manner.

In the literature, the centralized IPP problem has been extensively investigated, e.g., in [4], [5], [6]. Several works have also been dedicated to investigating the IPP problem in a distributed fashion. For instance, the author of [7] proposed the Kriged Kalman filter-based dynamic average consensus protocol to compute the common parameters for a spatial field model in each robot. It is noted that in a distributed scheme, each robot can only access its historical data and possibly the data locally exchanged with its neighbors, which it can use to build its own model of the spatial field. The robot then exploits its own spatial field model to not only predict the spatial phenomenon in the whole environment but also decide where to move in the next step. Still using the average consensus technique, the work in [2] presents a distributed Gaussian process (GP) learning algorithm for unmanned aerial vehicles to map an environmental quantity. In a recent work [8], the authors made use of the average consensus scheme to synchronize all local GP models in the robots. The motions of the robots were then proposed to be derived from a central unit, which is analogous to [9]. It can be observed that in distributed schemes, most of the existing works use the average consensus protocol to obtain common parameters for a local model in each robot, which can be better optimized. Furthermore, state-of-the-art methods for distributed IPP usually assume that the robot network is always connected; however, no specific attempts have been made by these methods to preserve the connectivity of the robot network. Without a connectivity-preserving mechanism, this assumption could be unrealistic, since robots may attempt to move far away from each other to gain more informative measurements.

To address the aforementioned shortcomings of existing methods, in this work, we propose to address the distributed

Manuscript received 22 August 2023; accepted 16 January 2024. Date of publication 5 February 2024; date of current version 15 February 2024. This letter was recommended for publication by Associate Editor G. Notomista and Editor M. Ani Hsieh upon evaluation of the reviewers' comments. This work was supported in part by the U.S. National Science Foundation under Grants NSF-CAREER: 1846513, NSF-PFI-TT: 1919127, and NSF-CAREER: 2238296 and in part by the Army Research Office under Grant W911NF-23-1-0186.
(Corresponding author: Thang Nguyen.)

Binh Nguyen and Thang Nguyen are with the Department of Engineering, Texas A&M University–Corpus Christi, Corpus Christi, TX 78412 USA (e-mail: binh.nguyen@tamucc.edu; thang.nguyen@tamucc.edu).

Truong X. Nghiem is with the School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011 USA (e-mail: truong.nghiem@nau.edu).

Linh Nguyen is with the Institute of Innovation, Science and Sustainability, Federation University Australia, Churchill, VIC 3842, Australia (e-mail: l.nguyen@federation.edu.au).

Hung M. La is with the Advanced Robotics and Automation Lab, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557 USA (e-mail: hla@unr.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3362133, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3362133

IPP problem while incorporating these issues into our design. Specifically, the robot network is required to actively maintain connectivity whilst the robots are exploring the environment. To this end, a distributed connectivity-preserving mechanism is proposed and incorporated into the planning optimization problem, which ensures that the network is always fully connected when the robots move to the next positions. In addition, the shared parameters of the local models of the spatial phenomenon in the robots are optimized instead of being computed by a simple average calculation. To solve the optimization problem in a distributed manner, a distributed optimization scheme is proposed based on a proximal alternating direction method of multiplier (proximal-ADMM). Our algorithm enables the network to not only optimally find the most informative next locations for the robots but also compute the optimal hyperparameters of each robot's spatial field model. We prove that the connectivity in the network is always maintained over time whilst the solution of the IPP optimization problem converges to a stationary point. We also verify the effectiveness of our proposed approach in synthetic experiments using a real-world dataset.

Notations: $\mathbb{N}$ is the set of natural numbers and $\mathbb{R}$ is the set of real numbers. $\mathbf{diag}(\cdot)$ stands for a block-diagonal matrix; $\otimes$ denotes the Kronecker product; $\mathbf{1}_n \in \mathbb{R}^n$ is a vector whose elements are 1. For an index set $\mathcal{S} = \{i_1, i_2, \ldots, i_n \,|\, i_j \in \mathbb{N}\}$, $[\mathcal{Q}_i]_{i\in\mathcal{S}} = [\, \mathcal{Q}_{i_1}^\top \quad \ldots \quad \mathcal{Q}_{i_n}^\top \,]^\top$ is the vertically concatenated matrix or vector of indexed real matrices of appropriate dimensions or scalars $\mathcal{Q}_i$. For a vector v, the notation $v \succ 0$ implies that all elements of v are greater than 0.

II. Distributed Modelling

A. Informative Model

This section presents the fundamental preliminary on a spatial field model used in the paper. Let us define a bounded convex set in ℓ-dimensional space $\mathcal{Q} \in \mathbb{R}^\ell$ as the space where the robots operate. The spatial field of interest is represented by a latent relationship $m : \mathcal{Q} \to \mathbb{R}$ that maps a robot location in $\mathcal{Q}$ to a spatial phenomenon at the location. At each time step t, the robot i takes a noisy measurement $y_{i,t} \in \mathbb{R}$ of the spatial field at its current location. In this paper, we assume that each robot has its own measurement model described as

$$y_{i,t} = m_i(\mathbf{p}_{i,t}) + \omega(\mathbf{p}_{i,t}), \tag{1}$$

where $\mathbf{p}_{i,t} \in \mathbb{R}^\ell$ is the robot location, $\omega(\mathbf{p}_{i,t}) \sim \mathcal{N}(0, \sigma_{\omega,i}^2)$ is an independent and identically distributed zero-mean Gaussian noise with standard deviation $\sigma_{\omega,i} > 0$, and $m_i(\mathbf{p}_{i,t}) \sim \mathcal{GP}(\mu_i, \kappa(\mathbf{p}_{i,t}, \mathbf{p}'_{i,t}))$ is a GP with covariance function $\kappa(\mathbf{p}_{i,t}, \mathbf{p}'_{i,t}) : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$ and a mean function μ_i. Let $\mathcal{D}_i = (\mathbf{p}_i, \mathbf{y}_i)$ be a training dataset of robot i including $D \in \mathbb{N}$ measurements $\mathbf{y}_i \in \mathbb{R}^D$ gathered at D positions $\mathbf{p}_i \in \mathbb{R}^{D\ell}$. At the locations of robot i over horizon $H \in \mathbb{N}$,

$$\hat{\bar{\mathbf{p}}}_i = \left[\hat{\mathbf{p}}_{i,t+h} \in \mathbb{R}^\ell\right]_{h\in\mathcal{H}} \; (\mathcal{H} = \{1, \ldots, H\}),$$

the predictions $\hat{\mathbf{y}}_{i,\mathcal{H}} = [\hat{y}_{i,t+h} \in \mathbb{R}]_{h\in\mathcal{H}}$ is a multivariate Gaussian distribution with the posterior covariance matrix

$$\Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_i} = K_{i,\hat{\bar{\mathbf{p}}}_i,\hat{\bar{\mathbf{p}}}_i} - K_{i,\hat{\bar{\mathbf{p}}}_i,\mathbf{p}_i}\left(K_{i,\mathbf{p}_i,\mathbf{p}_i} + \sigma_{\omega,i}^2 I\right)^{-1} K_{i,\mathbf{p}_i,\hat{\bar{\mathbf{p}}}_i} \tag{2}$$

where $K_{i,\hat{\bar{\mathbf{p}}}_i,\hat{\bar{\mathbf{p}}}_i} = \kappa_i(\hat{\bar{\mathbf{p}}}_i, \hat{\bar{\mathbf{p}}}_i) \in \mathbb{R}^{H\times H}$, $K_{i,\mathbf{p}_i,\mathbf{p}_i} = \kappa_i(\mathbf{p}_i, \mathbf{p}_i) \in \mathbb{R}^{D\times D}$, $K_{i,\hat{\bar{\mathbf{p}}}_i,\mathbf{p}_i} = K_{i,\mathbf{p}_i,\hat{\bar{\mathbf{p}}}_i}^\top = \kappa_i(\hat{\bar{\mathbf{p}}}_i, \mathbf{p}_i) \in \mathbb{R}^{H\times D}$, and I is a $D \times D$ identity matrix. It is noted that 2 will be used to design a distributed informative planner for the robot network.

B. Distributed Training

Consider M robots operating in $\mathcal{Q}$, where the communication space of each robot i at time t is assumed to be a sphere (or a circle in 2D) centered at $\mathbf{p}_{i,t}$ with radius R_i.

Definition 2.1: Robots i and j are said to be connected and neighbors of each other if they are in the communication range of each other, i.e. $\|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \leq \min(R_i, R_j)$.

Assumption 2.2: Each robot can exchange information with all of its neighbors.

Let $\mathcal{N}_{i,t} = \{j \in \mathcal{V} | \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \leq \min(R_i, R_j)\}$ define the set of neighbors of the robot i at step t, where $\mathcal{V} = \{1, 2, \ldots, M\}$. The vectors of positions and measurements that can be transferred to the robot i from its neighbors at time step t are indicated by $\mathbf{p}_{i,\mathcal{N}_{i,t}} = [[\mathbf{p}_{j,t}^\top, \tilde{\mathbf{p}}_{j,t-1}^\top]^\top]_{j\in\mathcal{N}_{i,t}}$ and $\mathbf{y}_{i,\mathcal{N}_{i,t}} = [[\mathbf{y}_{j,t}^\top, \tilde{\mathbf{y}}_{j,t-1}^\top]^\top]_{j\in\mathcal{N}_{i,t}}$, where $\tilde{\mathbf{p}}_{j,t-1}$ and $\tilde{\mathbf{y}}_{j,t-1}$ stand for vectors of historical (stored) positions and measurements in robot j but not in robot i up to time $t-1$. In addition, we define $\mathcal{D}_{i,t} = (\mathbf{p}_{i,\mathcal{D}}, \mathbf{y}_{i,\mathcal{D}})$ as the total data the robot i can have up to t, where $\mathbf{p}_{i,\mathcal{D}} = [\mathbf{p}_{i,t}^\top, \mathbf{p}_{i,\mathcal{N}_{i,k}}^\top]_{k=1,\ldots,t}^\top$ and $\mathbf{y}_{i,\mathcal{D}} = [\mathbf{y}_{i,t}^\top, \mathbf{y}_{i,\mathcal{N}_{i,k}}^\top]_{k=1,\ldots,t}$. To maintain a distributed manner in the network, we adopt the following assumption. At each time step t, each robot collects new data and uses them to improve the accuracy of the model by optimizing a cost function of the hyperparameter vector$\boldsymbol{\theta}_{i,t}$. The local hyperparameter can be obtained by optimizing a local log-marginal likelihood

$$\mathcal{L}_i(\boldsymbol{\theta}_{i,t}) = -\frac{1}{2}\mathbf{y}_{i,\mathcal{D}}^\top C_{\boldsymbol{\theta},i}^{-1}\mathbf{y}_{i,\mathcal{D}} - \text{logdet}\; C_{\boldsymbol{\theta},i}$$

where $C_{\boldsymbol{\theta},i} = \kappa(\mathbf{p}_{i,\mathcal{D}}, \mathbf{p}_{i,\mathcal{D}}) + \sigma_{\omega,i}^2 I$ is a matrix function of $\boldsymbol{\theta}_{i,t}$, $\kappa(\mathbf{p}_{i,\mathcal{D}}, \mathbf{p}_{i,\mathcal{D}})$ is chosen as a stationary covariance function [10]. Although this optimization can be implemented locally in each robot, the hyperparameters obtained in each robot are different. To alleviate the issue, the work in [8] presents a decentralized training formulated by the distributed optimization: $\min_{0\prec\boldsymbol{\theta}_{i,t}} \sum_{i=1}^M \mathcal{L}_i(\boldsymbol{\theta}_{i,t})$ subject to $\boldsymbol{\theta}_{i,t} = \boldsymbol{\theta}_{j,t} \forall i \in \mathcal{V} = \{1, 2, \ldots, M\}$, $j \in \mathcal{N}_{i,t}$. If the robots are connected while they move, the optimization problem can be solved by an inexact distributed consensus ADMM [11] or proximal-ADMM [12], which leads to the common hyperparameter values in all local GP models. However, ADMM-based algorithms need to solve suboptimizations repeatedly that directly increase computation and communication burdens. A less computationally expensive way is to minimize $\mathcal{L}_i(\boldsymbol{\theta}_{i,t})$, and dynamic average consensus [13] is used to synchronize all the local hyperparameters obtained.

III. Informative Path Planning With Connectivity Preservation

In this section, we first introduce multistep predictions based on informative path planning, which helps to find optimal sampling paths. Then, a mechanism is proposed to select important neighbors of a robot, which will be used to maintain the connectivity of the robot network. This will alleviate the constraints and computational costs. Lastly, the IPP optimization problem is formulated for the robot system subject to connectivity and collision avoidance constraints.

A. Multi-Step Predictions Based Informative Path Planning

IPP aims to find the next locations of the robots to maximize the information gain for the spatial field GP model. A commonly used optimal IPP method is maximizing the conditional entropy of the spatial field values at the next sampling locations given the current GP [4], [5]. The previous work [6] proposed a multi-step IPP method that looks ahead several steps to find optimal sampling paths, instead of finding only the next optimal locations.

Let $\hat{\bar{\mathbf{p}}}_i = (\hat{\mathbf{p}}_{i,t+1}, \ldots, \hat{\mathbf{p}}_{i,t+h}, \ldots, \hat{\mathbf{p}}_{i,t+H})$ stand for the optimal path the robot i computed at time step t where $\hat{\mathbf{p}}_{i,t+h} \in \mathbb{R}^\ell$ represents the predicted locations in the h step ahead and H denotes a predictive horizon. Note that the conditional entropy of the informative model with respect to the optimal path and dataset $\mathcal{D}_{i,t}$ up to current time t is given by logdet $\Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\hat{\bar{\mathbf{p}}}_i)$ [6], where logdet stands for the log-determinant function and $\Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\hat{\bar{\mathbf{p}}}_i)$ is defined by (2). Accordingly, the multi-step IPP is simply formulated by the following optimization problem

$$\underset{\hat{\mathbf{p}}_{i,t+1},\ldots,\hat{\mathbf{p}}_{i,t+H}\in\mathcal{Q}}{\text{maximize}} \ \text{logdet}\ \Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\hat{\bar{\mathbf{p}}}_i). \tag{3}$$

Note that the optimization problem (3) is for one robot. To distributively deploy a group of robots, we can modify (3) as maximizing a summation of all conditional entropy in each robot, i.e. maximize $\sum_{i=1}^{M}$ logdet $\Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\hat{\bar{\mathbf{p}}}_i)$. In addition, possible constraints for robot movement can be added, such as the maximum distance between two consecutive steps and collision avoidance.

B. Connectivity Preservation

This section presents a simple distributed mechanism to help each robot find necessary connections between its neighbors to preserve connections in the next step. Accordingly, if these connections can be maintained in the next step with regard to robot movements, the connectivity of the network graph induced by robots is preserved. To begin with, let us consider a model of robot movement described as

$$\mathbf{p}_{i,t+1} = \mathbf{p}_{i,t} + \mathbf{u}_{i,t} \tag{4}$$

where $\mathbf{u}_{i,t}$ stands for the displacement between two consecutive time steps, which is bounded as $\|\mathbf{u}_{i,t}\|_2 \le \delta_i$ with δ_i being the maximum distance robot i can travel in one-time step. To facilitate the construction of our mechanism to maintain the connectivity of the robot network, we introduce the following definition.

Definition 3.1: Robots i and j are said to be robustly connected at step t if $\|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \le \min(R_i, R_j) - \delta_i - \delta_j$.

Proposition 3.2: If robots i and j are robustly connected at t, then they will be connected at $t+1$ for any displacements $\|\mathbf{u}_{i,t}\|_2 \le \delta_i$ and $\|\mathbf{u}_{j,t}\|_2 \le \delta_j$.

Proof: From Definition 3.1, we have $\|\mathbf{p}_{i,t+1} - \mathbf{p}_{j,t+1}\|_2 \le \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 + \|\mathbf{u}_{i,t} - \mathbf{u}_{j,t}\|_2 \le \min(R_i, R_j)$. This completes the proof according to Definition 2.1. ∎

Proposition 3.2 guarantees the connectivity between robots i and j at the next time step under a certain condition. Here, we assume that communication ranges are significantly larger than the maximum traveled distances in one time step to ensure $\min(R_i, R_j) > \delta_i + \delta_j$ for all $i, j \in \mathcal{V}$.

Let $\mathcal{E}_t$ be the set of all (i,j), $i \ne j$, at time step t such that robots i and j are connected, i.e., $\mathcal{E}_t = \{(i,j) \in \mathcal{V}\times\mathcal{V} : \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \le \min(R_i, R_j)\}$. Denote $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ as an undirected graph of all robots, which may vary over time t, where $\mathcal{V} = \{1, 2, \ldots, M\}$ and $\mathcal{E}_t$ are the vertex and edge sets, respectively. The graph $\mathcal{G}_t$ is said to be connected if there exists at least a path from i to j for all $i \ne j$. In addition, let $\widetilde{\mathcal{G}}_{i,t+1} = (\mathring{\mathcal{N}}_{i,t}, \widetilde{\mathcal{E}}_{i,t+1})$ be the sub-graph defined by neighbors of robot i at step t and their connections at the next time step $t+1$ where $\mathring{\mathcal{N}}_{i,t} = \mathcal{N}_{i,t} \cup \{i\}$ and

$$\widetilde{\mathcal{E}}_{i,t+1} = \Big\{(v,n) \in \mathring{\mathcal{N}}_{i,t} \times \mathring{\mathcal{N}}_{i,t}, v \ne n \big| \quad \|\mathbf{p}_{v,t+1} - \mathbf{p}_{n,t+1}\|_2 \le \min(R_v, R_n)\Big\}.$$

Assumption 3.3: At the initial time $t = 0$, the group of robots is collision-free and the graph $\mathcal{G}_0$ is connected.

Lemma 3.4: Suppose that $\mathcal{G}_t$ is connected. If $\widetilde{\mathcal{G}}_{i,t+1}$ is connected for all $i \in \mathcal{V}$ then $\mathcal{G}_{t+1}$ is also connected.

Proof: This result is proved by contradiction. Assume that $\widetilde{\mathcal{G}}_{i,t+1}$ is connected and $\mathcal{G}_{t+1}$ is not connected. Then, at time step $t+1$, there exist two disjoint sets $\mathcal{S}_1$ and $\mathcal{S}_2$, where $\mathcal{S}_1 \cup \mathcal{S}_2 = \mathcal{V}$ and $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, such that there is no path between any pair of robots in $\mathcal{S}_1$ and $\mathcal{S}_2$. Since $\mathcal{G}_t$ is connected, there exist $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$ such that there is a connection between i and j at time t. Thus $j \in \mathcal{N}_{i,t}$. As $\widetilde{\mathcal{G}}_{i,t+1}$ is connected, there exists a path between i and j, which is a contradiction to the fact that $\mathcal{S}_1$ and $\mathcal{S}_2$ are disjoint. ∎

To simplify the presentation of our method, let us define two *connectivity-indicator* functions

$$c_t(i,j) = \begin{cases} 1 & \text{if } \min(R_i, R_j) - \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \ge 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$c_{\delta,t}(i,j) = \begin{cases} 1 & \text{if } \min(R_i, R_j) - \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 - \delta_i - \delta_j \ge 0, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $c_t(i,j) = 1$ indicates that robots i and j are *connected* at time t, i.e., $(i,j) \in \mathcal{E}_t$; and $c_{\delta,t}(i,j) = 1$ implies that robots i and j are *robustly connected* at time t, i.e., by Proposition 3.2, robots i and j are connected at both times t and $t+1$, regardless of their movements. When robot i and robot j are not neighbors, where robot i does not have the information $\mathbf{p}_{j,t}$ and δ_j of robot j, these connectivity indicator functions computed by robot i are

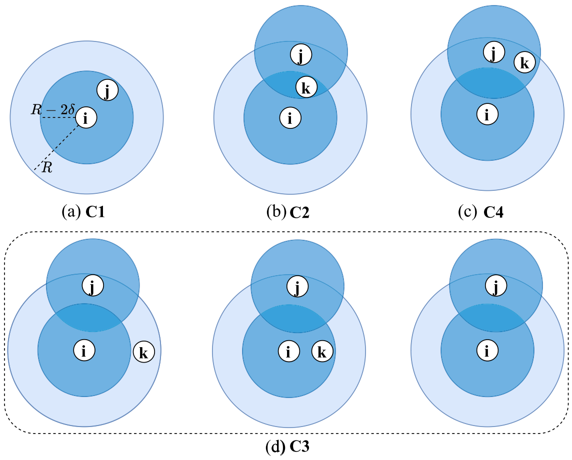


Fig. 1. Illustrations of the cases **C1**–**C4**. Here, $\delta_i = \delta_j = \delta, R_i = R_j = R$. The blue circles represent robust connection areas, that is, if robot j is inside a blue circle centered at robot i, they are *robustly connected*. **C1:** robots i and j are *robustly connected*. **C2:** robots i and j are not *robustly connected*, and robot k is *robustly connected* to robots i and j. **C3:** robots i and j are not *robustly connected*, and there does not exist a neighbor of robot i that is *robustly connected* to j (three possibilities are illustrated in Fig. 1(d)). **C4:** robots i and j are not *robustly connected*, and there exists a robot k that is *robustly connected* to robot j but not to robot i.

Algorithm 1: Distributed Connectivity Preservation.

```
Input Set of neighbors N_{i,t}, their positions p_{j,t}, communication radius R_i and δ_i.
Output Set of robots S_{i,t} (⊂ N_{i,t}) to be preserved.
Initiate: S_{i,t} = ∅
for j ∈ N_{i,t} do
  if c_{δ,t}(i, j) = 0 then                         ▷ C1
    Let S_{ij,t} = ∅
    for k ∈ N_{i,t} \ {j} do
      if c_{δ,t}(k, j) = 1 then
        if c_{δ,t}(i, k) = 0 then S_{ij,t} := S_{ij,t} ∪ {j}
        else Jump to 15                              ▷ C2
        end if
      end if
    end for
    if S_{ij,t} = ∅ then S_{i,t} := S_{i,t} ∪ {j}    ▷ C3
    else if j < min S_{ij,t} then
      S_{i,t} := S_{i,t} ∪ {j}                       ▷ C4
    end if
  end if
end for
```

set to 0. On the basis of these definitions, we have that $\mathcal{N}_{i,t} = \{j \in \mathcal{V} | c_t(i,j) = 1\}$. For a neighbor $j \in \mathcal{N}_{i,t}$ of robot i, let

$$\mathcal{S}_{ij,t} = \{k \in \mathcal{N}_{i,t} \backslash \{j\} | c_{\delta,t}(i,k) = 0 \wedge c_{\delta,t}(k,j) = 1\}.$$

be the set of neighbors of robots i and j that are *robustly connected* to j but not to i.

For a robot i at time t, there are four possible (disjoint) cases for any robot $j \in \mathcal{N}_{i,t}$:

C1 $c_{\delta,t}(i,j) = 1$,
C2 $c_{\delta,t}(i,j) = 0 \quad \wedge \quad (\exists k \in \mathcal{N}_{i,t} \setminus \{j\} : c_{\delta,t}(k,i) = 1 \wedge c_{\delta,t}(k,j) = 1)$,
C3 $c_{\delta,t}(i,j) = 0 \wedge (\nexists k \in \mathcal{N}_{i,t} \setminus \{j\} : c_{\delta,t}(k,j) = 1)$,
C4 $c_{\delta,t}(i,j) = 0 \wedge \mathcal{S}_{ij,t} \neq \emptyset$.

These cases are illustrated in Fig. 1 , where $\mathcal{S}_{ij,t} = \{\emptyset\}$ and $\mathcal{S}_{ij,t} = \{l, k\}$ in Fig. 1(c)–(d).

Based on these cases, let $\mathcal{S}_{i,t}$ be a set that represents neighbors of robot i determined at step t so that robot i attempts to maintain its connections with all robots in $\mathcal{S}_{i,t}$ at step $t+1$. Accordingly, the following actions are taken corresponding to each case from **C1** to **C4**. In **C1**, by *Proposition* 3.2, robots i and j are still connected at step $t+1$ for all admissible movements of robots i and j from step t to $t+1$, so we do not need to add j to $\mathcal{S}_{i,t}$. In **C2**, by the same arguments, there are connections (i,k) and (k,j) at step $t+1$, so j does not need to be added to $\mathcal{S}_{i,t}$. In **C3** and **C4**, we first determine the set $\mathcal{S}_{ij,t}$ at step t. If there does not exist a neighbor of robot $i (\neq j)$ *robustly connected* to j, robot j is added to $\mathcal{S}_{i,t}$. If $\mathcal{S}_{ij,t} \neq \emptyset$ (**C4**), robot j is added to $\mathcal{S}_{i,t}$ if $j < \min \mathcal{S}_{ij,t}$. We note here that the combination of **C2** and **C3** is a set of neighbor robots of robot i such that $\mathcal{S}_{ij,t} = \emptyset$. Based on these actions, Algorithm 1 is proposed to determine the set of connections to be preserved.

Remark 3.5: As mentioned in Proposition 3.2, two robustly connected robots are still connected in the next time step regardless of their movements. Thus, a robot does not have to maintain all its robust connections. Only selected neighbors of it are considered, provided that they satisfy certain conditions. Here, the introduction of $\mathcal{S}_{i,t}$, which identifies these neighbors, helps reduce the constraints and computational costs when solving the optimization problem introduced later. This is of significant importance when dealing with distributed computation for large-scale multi-agent systems.

The following lemma presents the main result of the distributed connectivity-preserving mechanism.

Lemma 3.6: Suppose that: (i) $\mathcal{G}_t$ is connected; (ii) no collision occurs in the group of robots; (iii) at time step $t+1$, robot i is connected with robots in $\mathcal{S}_{i,t}$ determined by Algorithm 1 for all $i \in \mathcal{V}$. Then $\mathcal{G}_{t+1}$ is connected.

Proof: Using Lemma 3.4, we only need to prove that $\widetilde{\mathcal{G}}_{i,t+1}$ is connected for all i, by showing that $\forall j \in \mathcal{N}_{i,t}$ there exists a path between j and i at the next step $t+1$. Consider each of the cases **C1**–**C4**. In **C1**, i.e., $c_{\delta,t}(i,j) = 1$, Proposition 3.2 gives $c_{t+1}(i,j) = 1$, implying that the connection (i,j) is maintained. In **C2**, because $c_{t+1}(i,k) = 1$ and $c_{t+1}(k,j) = 1$ for some $k \in \mathcal{N}_{i,t} \setminus j$, there is a path between i and j through k. In **C3**, the connection (i,j) is preserved at time $t+1$ by Algorithm 1. Finally, consider **C4**. Let $j' = \min_{k \in \mathcal{S}_{ij,t}} k$. Note that $j \notin \mathcal{S}_{ij,t}$. If $j < j'$, the connection (i,j) will be preserved by Algorithm 1 at time $t+1$. If $j' < j$, it has already $c_{\delta,t}(i,j) = 0$ and $c_{\delta,t}(j',j) = 1$, therefore, j' is also in **C4**, which implies that (j',j) is connected. By the same argument, either (i,j') is preserved by Algorithm 1 (i.e. $i \leftrightarrow j' \leftrightarrow j$) or (i,j') is not connected. If not connected, $\exists j'' : j'' = \min_{k \in \mathcal{S}_{ij',t}} k < j'$, which implies (j'',j')

is connected; hence, there is a path $j'' \leftrightarrow j' \leftrightarrow j$ where $j'' < j' < j$. Repetitively, either $(j'^{\cdots\prime}, i)$ is connected, or $(j'^{\cdots\prime}, i)$ is not connected at $t+1$ and $j'^{\cdots\prime} \leftrightarrow \cdots \leftrightarrow j' \leftrightarrow j$ where $j'^{\cdots\prime} < \ldots < j'' < j' < j$. Because $j'^{\cdots\prime}, \ldots, j'', j', j \in \mathcal{V}$ (a finite set), there always exists a path $i \leftrightarrow j'^{\cdots\prime} \leftrightarrow \cdots \leftrightarrow j' \leftrightarrow j$. ■

Remark 3.7: It should be noted that our proposed algorithm aims to preserve connectivity in the robot network, where the index order is used as discussed in Algorithm 1. Comparison of the index at line 13 of Algorithm 1 is a shorthand of comparison in other criteria, such as the distance traveled or the value of the cost function in the last step. In practice, we can use different criteria to obtain better navigation performance.

C. Optimization Formulation

The IPP problem with connectivity preservation can be formulated as an optimization problem described below. Here, each robot is represented by a circle with safety radius r_i centered at $\mathbf{p}_{i,t}$, for $i \in \mathcal{V}$. The collision avoidance constraint $\|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \geq r_i + r_j$ for each pair of robots i and j, where $j \in \mathcal{N}_{i,t}$, is non-convex for all t. The following result will allow us to rewrite the collision avoidance constraints as convex constraints.

Lemma 3.8 ([14]): For vector functions $f_1, f_2 : t \to \mathbb{R}^\ell$, if there exists a unit vector $\mathbf{e} : t \to \mathbb{R}^\ell (\|\mathbf{e}\|_2 = 1)$ such that

$$(f_1 - f_2)^\top \mathbf{e} \geq r_i + r_j, \tag{5}$$

then $\|f_1 - f_2\|_2 \geq r_i + r_j$ for all t.

With the help of Lemma 3.8, the collision avoidance condition between robots i and j ($j \in \mathcal{N}_{i,t}$) is given by:

$$\left(\hat{\mathbf{p}}_{i,t+h} - \hat{\mathbf{p}}_{j,t+h}\right)^\top \mathbf{e}_{ij,h} \geq r_i + r_j \tag{6}$$

where the unit vector $\mathbf{e}_{ij,h}$ is calculated from the current and predicted positions $\hat{\mathbf{p}}_{i,t-1+h|t-1}$ at the previous step $t-1$ as

$$\mathbf{e}_{ij,1} = (\mathbf{p}_{i,t} - \mathbf{p}_{j,t}) \left\|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\right\|_2^{-1}$$

and $\mathbf{e}_{ij,h} = (\hat{\mathbf{p}}_{i,t-1+h|t-1} - \hat{\mathbf{p}}_{j,t-1+h|t-1}) \left\|\hat{\mathbf{p}}_{i,t-1+h|t-1} - \hat{\mathbf{p}}_{j,t-1+h|t-1}\right\|_2^{-1}$ for $h = 2, \ldots, H$. Given the posterior covariance matrix $\Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}}$ in (2) and the total data $\mathcal{D}_{i,t}$, the IPP problem with collision avoidance and connectivity preservation can now be formulated as

$$\min_{\hat{\bar{\mathbf{p}}}_i} \quad -\sum_{i=1}^{M} \text{logdet } \Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\hat{\bar{\mathbf{p}}}_i) \tag{7a}$$

$$\text{s.t.} \quad \hat{\mathbf{p}}_{i,t+h} \in \mathcal{Q}, \tag{7b}$$

$$\|\hat{\mathbf{p}}_{i,t+h} - \hat{\mathbf{p}}_{i,t-1+h}\|_2 \leq \delta_i, \forall i \in \mathcal{V}, \tag{7c}$$

$$\|\hat{\mathbf{p}}_{i,t+h} - \hat{\mathbf{p}}_{j,t+h}\|_2 \leq \min(R_i, R_j), \forall j \in \mathcal{S}_{i,t}, \tag{7d}$$

$$\left(\hat{\mathbf{p}}_{i,t+h} - \hat{\mathbf{p}}_{j,t+h}\right)^\top \mathbf{e}_{ij,h} \geq r_i + r_j, \forall j \in \mathcal{N}_{i,t}, \tag{7e}$$

for all $h \in \mathcal{H}$ with $\hat{\mathbf{p}}_{i,t} = \mathbf{p}_{i,t}$. In the above optimization problem, (7b) stands for local constraints of limited movements in the operating area, (7c) captures the dynamics (4), the control input constraint (7d) is used to preserve connectivity between robot i and its neighbors in $\mathcal{S}_{i,t}$ in the next time step, and (7e) denotes the collision avoidance constraints. Because (7d) and (7e) are coupling constraints, it is challenging to solve the above optimization problem in a distributed manner. The next section will present a proximal-ADMM approach to overcome this challenge.

Theorem 3.9: Suppose that Assumption 3.3 holds an optimization problem (7a) is feasible for at all steps t. Then, the graph $\mathcal{G}_t$ is connected for all $t > 0$.

Proof: By induction, we assume that $\mathcal{G}_t$ is connected at step t. By the Algorithm 1, we can determine set $\mathcal{S}_{i,t}$. After solving (7a) at step t, robot i moves to $\hat{\mathbf{p}}_{i,t+1}$ at step $t+1$ for all $i \in \mathcal{V}$. The positions satisfy constraints (7d) and (7e), which implies that no collision occurs in a group of robots and connections between robot i and robots in set $\mathcal{S}_{i,t}$ are preserved. Thus, by Lemma 3.6, $\mathcal{G}_{t+1}$ is connected. ■

IV. Distributed Implementation

For each robot, let $\boldsymbol{\xi}_{ij,h} \in \mathbb{R}^\ell$ be a predicted (or virtual) position of robot j at predictive step $h \in \mathcal{H}$ calculated by robot i for all $j \in \mathcal{N}_{i,t}$. Indeed, robot i creates a replica $\boldsymbol{\xi}_{ij,h}$ of predicted positions of its neighbors $\hat{\mathbf{p}}_{j,h}$ to handle coupling constraints (7d) and (7e). For simplicity, we will omit the notation for dependence on t. Also to simplify notations, define $f_i(\hat{\bar{\mathbf{p}}}_i) = -\text{logdet } \Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\hat{\bar{\mathbf{p}}}_i), \boldsymbol{\xi}_{ij} = [\boldsymbol{\xi}_{ij,h}]_{h\in\mathcal{H}}$, $\boldsymbol{\xi}_i = [\boldsymbol{\xi}_{ij}]_{j\in\mathring{\mathcal{N}}_{i,t}}$, and time-varying convex sets

$$\mathcal{C}_i = \Big\{\mathbf{z} = [\mathbf{z}_j]_{j\in\mathring{\mathcal{N}}_{i,t}}, \mathbf{z}_j = \left[\mathbf{z}_{j,h} \in \mathbb{R}^\ell\right]_{h=1,\ldots,H} \mid \mathbf{z}_{i,h} \in \mathcal{Q}, \|\mathbf{z}_{i,h+1} - \mathbf{z}_{i,h}\|_2 \leq \delta_i, \|\mathbf{z}_{i,h} - \mathbf{z}_{j,h}\|_2 \leq \min(R_i, R_j), j \in \mathcal{S}_{i,t}, (\mathbf{z}_{i,h} - \mathbf{z}_{j,h})^\top \mathbf{e}_{ij,h} \geq r_i + r_j, j \in \mathcal{N}_{i,t}.\Big\}$$

for $i \in \mathcal{V}$ representing for local constraints. By Assumption 3.3, the initial position $[\mathbf{1}_H \otimes \mathbf{p}_{j,1}]_{j\in\mathring{\mathcal{N}}_{i,1}} \in \mathcal{C}_i$ for all i. As a result, $\mathcal{C}_i$ is a nonempty set at the initial time step. Thus, (7a) is feasible at the initial time step, and then $[\mathbf{1}_H \otimes \mathbf{p}_{j,2}]_{j\in\mathring{\mathcal{N}}_{i,2}} \in \mathcal{C}_i$ at $t = 2$. Repeatedly, for the next steps, $\mathcal{C}_i$ is also nonempty. It is straightforward to see that $\mathcal{C}_i$ is a closed convex set by its definition. Here, the use of $\mathcal{C}_i$ with indicator function $\mathcal{I}_{\mathcal{C}_i}(\bullet)$ of $\mathcal{C}_i$ ($\mathcal{I}_{\mathcal{C}_i}(x) = 0$ if $x \in \mathcal{C}_i$, $\mathcal{I}_{\mathcal{C}_i}(x) = \infty$ else) is for converting optimization (7a) into the following consensus form:

$$\min_{\hat{\bar{\mathbf{p}}}_i} \sum_{i=1}^{M} f_i(\hat{\bar{\mathbf{p}}}_i) + \mathcal{I}_{\mathcal{C}_i}(\boldsymbol{\xi}_i), \text{ s.t. } \hat{\bar{\mathbf{p}}}_i - \boldsymbol{\xi}_{ji} = 0, j \in \mathring{\mathcal{N}}_{i,t}. \tag{8}$$

The optimization (8) can be solved by the nonconvex ADMM in [15] with global convergence proofs. However, solving the nonconvex problems repeatably results in many heavy computational costs for each robot even in parallel fashions. Thus, by taking advantage of proximal ADMM [12], we take the first-order proximal approximation of $f_i(\hat{\bar{\mathbf{p}}}_i)$ at $\hat{\bar{\mathbf{p}}}_i^{(k)}$ as $\bar{f}_i(\hat{\bar{\mathbf{p}}}_i; \hat{\bar{\mathbf{p}}}_i^{(k)}) = f_i(\hat{\bar{\mathbf{p}}}_i^{(k)}) + \nabla f_i^\top(\hat{\bar{\mathbf{p}}}_i^{(k)})(\hat{\bar{\mathbf{p}}}_i - \hat{\bar{\mathbf{p}}}_i^{(k)}) + \frac{L_i}{2}\|\hat{\bar{\mathbf{p}}}_i - \hat{\bar{\mathbf{p}}}_i^{(k)}\|_2^2$ where $L_i > 0$ and $\hat{\bar{\mathbf{p}}}_i^{(k)}$ is the unique solution at iteration k. Moreover, let us define an augmented Lagrangian function: $\mathcal{L} = \sum_{i=1}^{M} \mathcal{L}_i(\hat{\bar{\mathbf{p}}}_i, \boldsymbol{\lambda}_{ij}, \boldsymbol{\xi}_{ji})$, where $\mathcal{L}_i = \bar{f}_i(\hat{\bar{\mathbf{p}}}_i; \hat{\bar{\mathbf{p}}}_i^{(k)}) + \mathcal{I}_{\mathcal{C}_i}(\boldsymbol{\xi}_i) + \sum_{j\in\mathring{\mathcal{N}}_{i,t}} \boldsymbol{\lambda}_{ij}^\top(\hat{\bar{\mathbf{p}}}_i - \boldsymbol{\xi}_{ji}) + \frac{\varrho}{2}\|\hat{\bar{\mathbf{p}}}_i - \boldsymbol{\xi}_{ji}\|_2^2$,

$\boldsymbol{\lambda}_{ij}$ is vector of the dual variable, and $\rho > 0$ is a regularization parameter. Let $\boldsymbol{\xi}$ and $\hat{\mathbf{\mathring{p}}}$ be vectors of $\boldsymbol{\xi}_i$ and $\hat{\mathbf{\mathring{p}}}_i$, then ADMM for (8) is

$$\boldsymbol{\xi}^{(k+1)} = \arg\min \mathcal{L}\left(\hat{\mathbf{\mathring{p}}}^{(k)}, \boldsymbol{\xi}, \boldsymbol{\lambda}^{(k)}\right), \quad (9a)$$

$$\hat{\mathbf{\mathring{p}}}^{(k+1)} = \arg\min \mathcal{L}\left(\hat{\mathbf{\mathring{p}}}, \boldsymbol{\xi}^{(k+1)}, \boldsymbol{\lambda}^{(k)}\right), \quad (9b)$$

$$\boldsymbol{\lambda}_{ij}^{(k+1)} = \boldsymbol{\lambda}_{ij}^{(k)} + \rho\left(\hat{\mathbf{\mathring{p}}}_i^{(k+1)} - \boldsymbol{\xi}_{ji}^{(k+1)}\right) \quad (9c)$$

where $\boldsymbol{\xi} = [\boldsymbol{\xi}_i]_{i\in\mathcal{V}}$, $\hat{\mathbf{\mathring{p}}} = [\hat{\mathbf{\mathring{p}}}_i]_{i\in\mathcal{V}}$. By rearranging $\mathcal{L}(\hat{\mathbf{\mathring{p}}}^{(k)}, \boldsymbol{\xi}, \boldsymbol{\lambda}^{(k)})$, (9a) can be computed in parallel by the constrained quadratic programming problem:

$$\boldsymbol{\xi}_i^{(k+1)} = \arg\min \sum_{j\in\mathring{\mathcal{N}}_{i,t}} \left\| \hat{\mathbf{\mathring{p}}}_j^{(k)} - \boldsymbol{\xi}_{ij} + \frac{1}{\rho}\boldsymbol{\lambda}_{ji}^{(k)} \right\|_2^2 + \mathcal{I}_{\mathcal{C}_i}(\boldsymbol{\xi}_i). \quad (10)$$

Analogously, (9b) can be split into unconstrained convex quadratic programming problems yielding:

$$\hat{\mathbf{\mathring{p}}}_i^{(k+1)} = \frac{\sum_{j\in\mathring{\mathcal{N}}_{i,t}}\left(\rho\boldsymbol{\xi}_{ji}^{(k+1)} - \boldsymbol{\lambda}_{ij}^{(k)}\right) - \nabla f_i(\hat{\mathbf{\mathring{p}}}_i^{(k)}) + L_i\hat{\mathbf{\mathring{p}}}_i^{(k+1)}}{L_i + \rho|\mathring{\mathcal{N}}_{i,t}|} \quad (11)$$

where $|\mathring{\mathcal{N}}_{i,t}|$ is the number of elements in $\mathring{\mathcal{N}}_{i,t}$. The iteration will stop when all residual errors $\sum_{j\in\mathring{\mathcal{N}}_i} \|\boldsymbol{\xi}_{ij}^{(k+1)} - \hat{\mathbf{\mathring{p}}}_j^{(k+1)}\|_2$ are smaller than a given threshold $\epsilon_{\text{res}} > 0$. Then, (10), (11) and (9c) boil down to Algorithm 2.

Remark 4.1: Algorithm 1 and Algorithm 2 represent the proposed distributed IPP method. These are followed after completing distributed GP training in Section II-B. The data set $\mathcal{D}_{i,t}$ and the hyperparameter $\boldsymbol{\theta}_{i,t}$ of each robot are only updated in the GP training process. Therefore, the convergence of Algorithm 2 at each time step is not affected by training the GP model.

To prove the convergence of Algorithm 2, this paper adopts the squared exponential covariance function: $\kappa_i(\mathbf{p}_i, \mathbf{p}'_i) = \sigma_{\kappa,i}^2 \exp(-(\mathbf{p}_i - \mathbf{p}'_i)^\top \Phi_i^{-2}(\mathbf{p}_i - \mathbf{p}'_i))$ with $\Phi_i = \mathbf{diag}(\phi_{i,1}, \ldots, \phi_{i,\ell})$ and a hyperparameter vector $\boldsymbol{\theta}_i = [\phi_{i,1}^{-2}, \ldots, \phi_{i,\ell}^{-2}, \sigma_{\kappa,i}^2]^\top$. For other covariance functions such as linear, rational quadratic, and exponential functions whose second derivatives are continuous, proofs of convergence are derived in the same manner. It is stressed that (9a)–(9c) are analogous to the proximal-ADMM [12, Remark 1]. Thus, we inherit the conditions in [12] to prove the convergence.

Proposition 4.2: $\nabla f_i(.)$ is Lipschitz continuous.

Proof: According to (2), for vector $\mathbf{s} = [\mathbf{s}_h]_{h\in\mathcal{H}} \in \mathcal{Q}^H$ where $\mathbf{s}_h = [s_{hl}]_{1\le l\le\ell}$, the gradient $\nabla f_i(\mathbf{s}) = [\frac{\partial f_i}{\partial s_{11}}, \ldots, \frac{\partial f_i}{s_{1\ell}}, \ldots, \frac{\partial f_i}{s_{H1}}, \ldots, \frac{\partial f_i}{s_{H\ell}}]^\top$ is given by $\frac{\partial f_i}{\partial s_{hl}} = 2\mathbf{Tr}(\Sigma_{\hat{\mathbf{y}}_{i,\mathcal{H}}|\mathcal{D}_{i,t}}(\mathbf{s}) K_{i,\mathbf{s},\mathbf{p}}(K_{i,\mathbf{p},\mathbf{p}} + \sigma_{i,\omega}^2 I)\frac{\partial K_{i,\mathbf{s},\mathbf{p}}^\top}{\partial s_{hl}})$ where $K_{i,\mathbf{s},\mathbf{p}} = \kappa_i(\mathbf{s}, \mathbf{p}_{i,\mathcal{D}})$ and $\mathbf{p}_{i,\mathcal{D}}$ is given in Section II-B. Note that for vectors $\mathbf{s}_h$ and $\mathbf{p} = [p_l]_{1\le l\le\ell} \in \mathcal{Q}$, $\frac{\partial\kappa_i(\mathbf{s}_h,\mathbf{p})}{\partial s_{hl}} = \phi_{i,n}^{-2}(p_l - s_{hl})\kappa_i(\mathbf{s}_h, \mathbf{p})$. Hence, $|\frac{\partial\kappa_i(\mathbf{s}_h,\mathbf{p})}{\partial s_{hl}}| \le \sigma_{i,\kappa}^2\phi_{i,n}^{-2}|p_l - s_{hl}|$, then $\frac{\partial f_i}{\partial s_{hl}}$ and $\nabla f_i(\mathbf{s})$ are bounded. In the same argument, taking partial derivatives $\frac{\partial^2\kappa_i(\mathbf{s}_h,\mathbf{p})}{\partial s_{hl}^2}$ and $\frac{\partial^2\kappa_i(\mathbf{s}_h,\mathbf{p})}{\partial s_{hl}\partial s_{hn}} (l \ne n)$, we can show that these derivatives are continuous. Therefore, as a sum of $\frac{\partial^2\kappa_i(\mathbf{s}_h,\mathbf{p})}{\partial s_{hl}^2}$, $\frac{\partial^2\kappa_i(\mathbf{s}_h,\mathbf{p})}{\partial s_{hl}\partial s_{hn}}$ and $\frac{\partial f_i}{\partial s_{hl}}$, $\frac{\partial^2 f_i}{\partial s_{hl}\partial s_{hl}}$ and $\frac{\partial^2 f_i}{\partial s_{hl}\partial s_{hn}}$ is also continuous. Hence, $\nabla f_i(.)$ is continuously differentiable. ∎

Algorithm 2: Proximal ADMM-Based Algorithm.

Input: M, R_i, r_i, ρ, residual threshold ϵ_{res} and maximum iteration $N_{\max}$
Output: $\hat{\mathbf{\mathring{p}}}_i = [\hat{\mathbf{p}}_{i,t+h}]_{h\in\mathcal{H}}$
Initiate: $\hat{\mathbf{\mathring{p}}}_i^{(1)} = \mathbf{1}_H \otimes \mathbf{p}_{i,0}$ for $h \in \mathcal{H}$, $\boldsymbol{\lambda}_{ij}^{(1)} = 0$
for $k = 1, 2, \ldots, N_{\max}$ **do**
  Robot i sends $\hat{\mathbf{\mathring{p}}}_i^{(k)} + \frac{1}{\rho}\boldsymbol{\lambda}_{ij}^{(k)}$ to robot j and receives $\hat{\mathbf{\mathring{p}}}_j^{(k)} + \frac{1}{\rho}\boldsymbol{\lambda}_{ji}^{(k)}$ from robot $j \in \mathcal{N}_{i,t}$
  Solve (10) to obtain $\boldsymbol{\xi}_{ij}^{(k+1)}$
  Robot i sends $\boldsymbol{\xi}_{ij}^{(k)}$ to robot j and receives $\boldsymbol{\xi}_{ji}^{(k)}$ from robot $j \in \mathcal{N}_{i,t}$
  Calculate $\hat{\mathbf{\mathring{p}}}_i^{(k+1)}$ by (11)
  if $\max_{i\in\mathcal{V}} \sum_{j\in\mathring{\mathcal{N}}_i} \|\boldsymbol{\xi}_{ij}^{(k+1)} - \hat{\mathbf{\mathring{p}}}_j^{(k+1)}\|_2 < \epsilon_{\text{res}}$ **then**
    return $\hat{\mathbf{\mathring{p}}}_i^{(k+1)}$
  else Calculate $\boldsymbol{\lambda}_{ij}^{(k+1)}$ by (9c).
  end if
end for

Proposition 4.3: Suppose that Assumption 3.3 holds, the sequence $\{(\boldsymbol{\xi}^{(k)}, \hat{\mathbf{\mathring{p}}}^{(k)}, \boldsymbol{\lambda}^{(k)})\}_{k=0,1,\ldots}$ generated by Algorithm 2 converges to a stationary point $(\boldsymbol{\xi}^\star, \hat{\mathbf{\mathring{p}}}^\star, \boldsymbol{\lambda}^\star)$, and $\sum_{i\in\mathcal{V}} f_i(\hat{\mathbf{\mathring{p}}}_i^\star) < \sum_{i\in\mathcal{V}} f_i(\hat{\mathbf{\mathring{p}}}_i^{(0)})$.

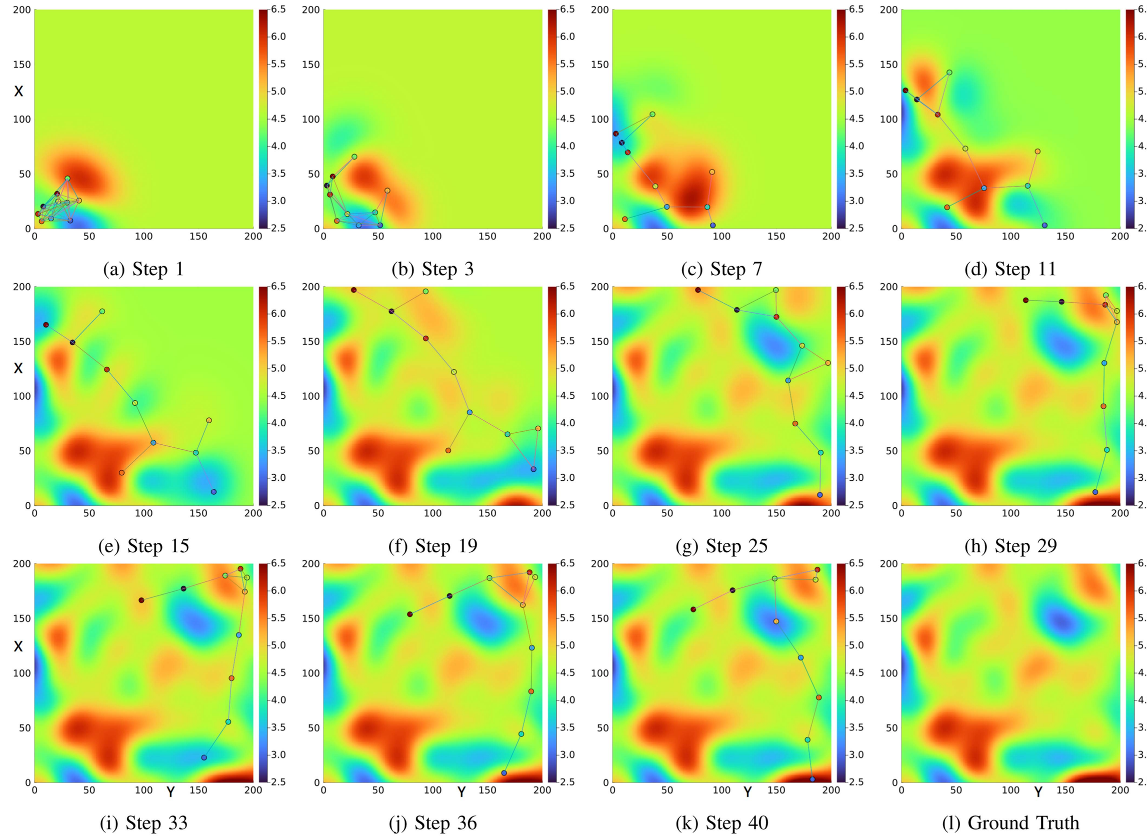
Proof: By Proposition 4.2, (8) is a special case of the problem addressed in [12] with a non-smooth part $\mathcal{I}_{\mathcal{C}_i}(\boldsymbol{\xi}_i)$. Thus, the proof of this proposition follows from [12, Theorem 2]. ∎

For other covariance functions where the second derivative covariance functions do not exist, the convergence proof of the proximal-ADMM can be deduced from [15].

V. Numerical Results

To demonstrate the effectiveness of the proposed approach, we conducted synthetic experiments by using the real-world dataset of the soil organic matter (SOM) in [4], [16]. The dataset comprises 1375 data points. For the purposes of evaluation, we utilized all the datasets to build a model, which was then employed to create a ground truth of the spatial field, e.g. SOM in the whole environment. A map of the ground truth is illustrated in Fig. 2(l). Afterward, we used the ground truth to verify the accuracy of the predicted results obtained by our planning algorithms in each robot.

In a nutshell, to informatively explore the spatial field, each robot in the network was expected to move to a new location and take a new measurement at each time step. In our simulations, we assumed that the robots measured SOM in the previously created ground truth map. Our IPP method aimed to drive the robots to the most informative locations in the environment given their constraints of connectivity, collision avoidance, and maximum travel distance allowance. Eventually, each robot only utilized its historical data and exchanged it locally with its neighbors to distributely learn its own GP model of the SOM in the entire



(a) Step 1 (b) Step 3 (c) Step 7 (d) Step 11
(e) Step 15 (f) Step 19 (g) Step 25 (h) Step 29
(i) Step 33 (j) Step 36 (k) Step 40 (l) Ground Truth

Fig. 2. Snapshots of all the robot movements at some specific time steps are demonstrated in (a)–(k). The corresponding spatial field in the whole environment predicted by the robot 1 (black) is also shown in (a)–(h), which, over time, is highly comparable with the ground truth in (i).

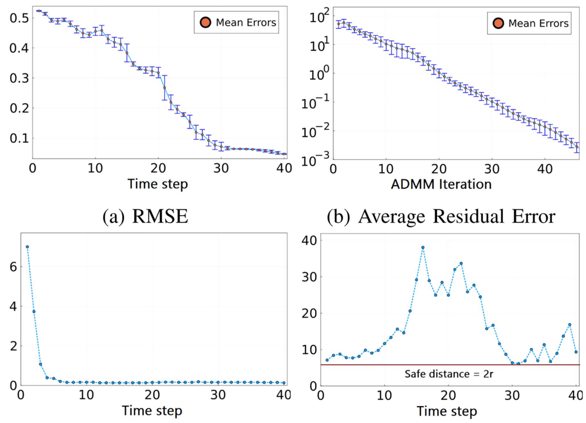
environment. Although each robot was expected to navigate through a limited number of locations, it could still predict the spatial field in the whole space, similar to the ground truth.

Our distributed GP training process is performed in two steps: (i) local training and (ii) consensus in hyperparameter $\boldsymbol{\theta}_{i,t}$. In local training, each robot optimizes a local logarithmic marginal likelihood $\mathcal{L}_i(\boldsymbol{\theta}_{i,t})$ to obtain its local hyperparameters. The complexity of this step is $\mathcal{O}(n_{\mathcal{D}}^3)$, where $n_{\mathcal{D}}$ is the number of data points, but can be reduced to $\mathcal{O}(n_{\mathcal{D}}m^2)$ by techniques such as *subset of datapoints* and *spare GP*, where $0 < m < n_{\mathcal{D}}$ is the size of the data subset or the inducing points. Here, we used the *subset of datapoints* technique [10] to alleviate computational loads. The consensus algorithm is used to average the local hyperparameters to obtain a common hyperparameter. The algorithm is scalable for a large number of robots, and its complexity in the worst case is $\mathcal{O}(M^2)$. We evaluated the computational loads of the proposed IPP method on a computer with an i5-8700 CPU and 16 Gb RAM. Of the ten experiments that we conducted, the average computational times at time steps 10, 20, 30, and 40 are 2.8[s], 4.4[s], 10.5[s], and 16.3[s], respectively.

In the experiments, we used $M = 10$ mobile robots operating in a space of $200[m] \times 200[m]$. Each robot with a safety boundary was represented by a circle with a radius of $r = 3[m]$ while its communication range was defined by $R = 40[m]$. At each time step, a robot was allowed to travel up to $\delta_i = 10[m]$. To run the proposed algorithms, we set $H = 3$, $N_{\max} = 100$, $L_i = 0.1$ and $\epsilon_{\text{res}} = 0.01$. Moreover, to preserve connectivity in the network, we implemented the cases **C1-C4**, where the cost function value was used in **C4** as mentioned in Remark 3.7. The horizon length H is the number of predictive steps, and a larger H leads to better navigation in areas with high entropy, far from the robot's current position. However, a long horizon length causes computational and time requirements between two consecutive steps. Thus, depending on the specific application, the horizon length H can be chosen appropriately to compromise between the accuracy of the GP model and the computational burden.

The regularization parameter ρ was defined in a strictly increasing sequence $\rho_{k+1} = 1.2\rho_k$ so that the convergence of Algorithm 2 was accelerated. The starting position of each robot was randomly selected in a corner of the space provided that the initial graph of the robot network was connected. The experiments were implemented in Julia [17] using the *Gaussian Processes* package [18]. The code, data, videos, and other results of the experiments can be found at https://github.com/AACLab/DSTB-INFO-PP-GP.

To illustrate the results obtained, we took several snapshots of all the robot movements over time and demonstrated them in Fig. 2(a)–(k). It is clearly seen that there was no collision among the robots while they were maintained to be connected to at least one of their neighbors over time. All robots were driven to the most informative locations within their maximum travel distance allowance. To demonstrate that each robot could predict well the



(a) RMSE (b) Average Residual Error

(c) Fiedler eigenvalue (d) Minimum distance

Fig. 3. (a) RMSE, (b) average residual errors of ADMM over iterations, (c) proof of connectivity preservation, and (d) proof of no collision: the minimum distance between any pair of robots is always positive.

spatial field in a distributed manner given its limited dataset at each time step, we employed the GP model learned by robot 1 and predicted the SOM throughout the space. The prediction results are also shown in Fig. 2(a)–(k). It can be seen that over time the predictions gradually approached the ground truth, as shown in Fig. 2(l). After 40 movements, the SOM estimation in the whole environment is highly comparable to the ground truth.

To quantify the accuracy of the predictions obtained by each robot, we computed the mean square errors (RMSE) between the estimation and the ground truth at each time step. Here, 10 errors in 10 robots at a time step are summarized by a mean and a standard deviation. And those over time are plotted in Fig. 3(a). As can be seen, the more data is collected, the smaller the errors in each robot. At time step 40, the prediction errors in all the robots are close to zero. We also illustrate the convergence of the proximal ADMM algorithm used in our planners for 10 robots in Fig. 3(b). It can be seen that the average residual errors were significantly reduced over iterations when the algorithm was run, which proved that the solution of the optimization problem converged to a stationary point. On the other hand, we calculated the Fiedler eigenvalue (the second-smallest eigenvalue of the Laplace matrix induced by the robot network) to demonstrate the preservation of connectivity in the robot network. It is noted from [19] that at a particular time if the Fiedler eigenvalue is positive, all robots are fully connected. In our experiments, the resulting connectivity-indicator functions as depicted in Fig. 3(c) demonstrate that the connectivity of the network was maintained while the robots explored the environment. In addition, Fig. 3(d) shows the minimum distance between two agents which demonstrates that there is no collision in the robot group.

VI. Concluding Remarks

The paper has presented an efficient distributed IPP approach for a mobile robot network to explore and simultaneously predict a spatial environmental field of interest using a GP model in each robot, whose connectivity is fully maintained throughout its exploration time. The IPP problem is informatively formulated by using the multistep prediction scheme given the communication and collision avoidance constraints, which is effectively solved by employing the proximal ADMM algorithm in a distributed manner. The effectiveness of the proposed method was not only theoretically proved, but also promisingly verified in simulations using the real-world dataset. Our future work will aim to validate our technique in real robots in cluttered environments where communication areas are affected by obstacles. In addition, noise and uncertainties in robotic systems will be considered.

References

[1] J. Kim, "Coverage control of multiple robots in cluttered three-dimensional environments," *IET Radar Sonar Navigation*, vol. 15, no. 9, pp. 1016–1029, 2021.
[2] D. Jang, J. Yoo, C. Y. Son, D. Kim, and H. J. Kim, "Multi-robot active sensing and environmental model learning with distributed Gaussian process," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5905–5912, Oct. 2020.
[3] C. Van Nguyen and H.-S. Ahn, "Distributed solving exact potential games via differential inclusions and consensus algorithms," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 4212–4217.
[4] L. Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake, "Mobile robotic sensors for environmental monitoring using Gaussian Markov random field," *Robotica*, vol. 39, no. 5, pp. 862–884, 2021.
[5] L. V. Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake, "Information-driven adaptive sampling strategy for mobile robotic wireless sensor network," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 372–379, Jan. 2016.
[6] V.-A. Le, L. Nguyen, and T. X. Nghiem, "Multi-step predictions for adaptive sampling in mobile robotic sensor networks using proximal ADMM," *IEEE Access*, vol. 10, pp. 64850–64861, 2022.
[7] J. Cortés, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Trans. Automat. Control*, vol. 54, no. 12, pp. 2816–2827, Dec. 2009.
[8] G. P. Kontoudis, "Communication-aware, scalable Gaussian processes for decentralized exploration," Ph.D. dissertation, Virginia Tech, Blacksburg, VA, USA, 2022.
[9] V.-A. Le and T. X. Nghiem, "Distributed experiment design and control for multi-agent systems with Gaussian processes," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 2226–2231.
[10] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.
[11] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.
[12] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.
[13] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, 2009.
[14] B. Nguyen et al., "Distributed formation trajectory planning for multi-vehicle systems," in *Proc. Amer. Control Conf.*, 2023, pp. 1325–1330.
[15] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.
[16] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016.
[17] A. Sengupta, *Julia High Performance: Optimizations, Distributed Computing, Multithreading, and GPU Programming With Julia 1.0 and Beyond*. Birmingham, U.K.: Packt Publishing Ltd, 2019.
[18] J. Fairbrother, C. Nemeth, M. Rischard, J. Brea, and T. Pinder, "GaussianProcesses: Jl: A nonparametric Bayes package for the Julia language," *J. Stat. Softw.*, vol. 102, pp. 1–36, 2022.
[19] A. Simonetto, T. Keviczky, and R. Babuška, "Constrained distributed algebraic connectivity maximization in robotic networks," *Automatica*, vol. 49, no. 5, pp. 1348–1357, 2013.