



# Multi-USV Adaptive Exploration Using Kernel Information and Residual Variance

Rajat Mishra<sup>1\*</sup>, Teong Beng Koay<sup>1,2</sup>, Mandar Chitre<sup>1,3</sup> and Sanjay Swarup<sup>2,4,5</sup>

<sup>1</sup> Acoustic Research Laboratory, Tropical Marine Science Institute, National University of Singapore, Singapore, Singapore, <sup>2</sup> NUS Environmental Research Institute, National University of Singapore, Singapore, Singapore, <sup>3</sup> Department of Electrical & Computer Engineering, Faculty of Engineering, National University of Singapore, Singapore, Singapore, <sup>4</sup> Singapore Centre for Environmental Life Sciences Engineering, Singapore, Singapore, <sup>5</sup> Department of Biological Sciences, Faculty of Science, National University of Singapore, Singapore, Singapore

Using a team of robots for estimating scalar environmental fields is an emerging approach. The aim of such an approach is to reduce the mission time for collecting informative data as compared to a single robot. However, increasing the number of robots requires coordination and efficient use of the mission time to provide a good approximation of the scalar field. We suggest an online multi-robot framework *m-AdaPP* to handle this coordination. We test our framework for estimating a scalar environmental field with no prior information and benchmark the performance via field experiments against conventional approaches such as lawn mower patterns. We demonstrated that our framework is capable of handling a team of robots for estimating a scalar field and outperforms conventional approaches used for approximating water quality parameters. The suggested framework can be used for estimating other scalar functions such as air temperature or vegetative index using land or aerial robots as well. Finally, we show an example use case of our adaptive algorithm in a scientific study for understanding micro-level interactions.

**Keywords:** multi-robot systems, informative path planning, Gaussian process, field validated, sampling hotspots, freshwater analysis

## OPEN ACCESS

### Edited by:

Victor Zykov,  
Independent Researcher, Menlo Park,  
CA, United States

### Reviewed by:

Francesco Amigoni,  
Politecnico di Milano, Italy  
Nils Einecke,  
Honda Research Institute Europe  
GmbH, Germany

### \*Correspondence:

Rajat Mishra  
rajat.mishra@u.nus.edu

### Specialty section:

This article was submitted to  
Multi-Robot Systems,  
a section of the journal  
Frontiers in Robotics and AI

**Received:** 13 June 2020

**Accepted:** 25 March 2021

**Published:** 28 May 2021

### Citation:

Mishra R, Koay TB, Chitre M and  
Swarup S (2021) Multi-USV Adaptive  
Exploration Using Kernel Information  
and Residual Variance.  
Front. Robot. AI 8:572243.  
doi: 10.3389/frobt.2021.572243

## 1. FRAMEWORKS FOR ENVIRONMENTAL MONITORING

### 1.1. Current Practices in Environmental Monitoring

Environmental processes often exhibit large scale features, generally in the range of kilometers, and vary both spatially and temporally. In order to monitor these processes through environmental parameters such as pH or dissolved oxygen (DO), it is ideal to have multi-fold coverage of the survey area. Buoys and floats equipped with environmental sensors are used to monitor water quality across different water resources such as oceans and freshwater systems. One of the widely used platforms is Argo Floats (Roemmich et al., 1999), which has helped in various scientific studies (Siswanto et al., 2008; Hosoda et al., 2009; Mignot et al., 2014; Stanev et al., 2014). A common approach is to place static buoys based on prior information from environmental modeling (Krause et al., 2008; Hart and Murray, 2010). Such an approach provides good temporal resolution, however, it is resource intensive as each buoy requires environmental sensors and regular maintenance.

More recently, robotic systems such as autonomous underwater vehicles (AUVs) and unmanned surface vehicles (USVs) are being increasingly used as fundamental data-gathering tools by

scientists, catering to the need of environmental monitoring and sampling (Dunbabin and Marques, 2012). A large fraction of AUVs today are designed to carry out scientific data collection missions (Pascoal et al., 2000; Sukhatme et al., 2007; Zhang et al., 2012; Hitz et al., 2014; Koay et al., 2015). We also developed such robotic systems for water quality monitoring as shown in **Figure 1B**. Such robot-aided data collection has been also used to explain biological processes (Caron et al., 2008; Camilli et al., 2010). However, the use of these robots is still limited due to the complex spatio-temporal nature of the environmental parameters. Adaptive planning frameworks such as Informative Path Planning (IPP) are generally used to overcome such limitations and perform environmental monitoring missions (Zhang and Sukhatme, 2007; Smith et al., 2011; Cao et al., 2013; Hitz et al., 2017).

One of the challenges in using adaptive planning frameworks is the data collection process. In general, the IPP framework mitigates this challenge by evaluating paths using an informative criterion for unobserved locations (Sukhatme et al., 2007; Low et al., 2008, 2011; Yu et al., 2016; Ma et al., 2017), shown as an illustration in **Figure 1A**. The robot then traverses the path that provides maximum information as per the predefined criterion and collects data to give an estimate of the environment. In general, the IPP frameworks have three components: collecting data while traversing, adapting the robot's path to provide a good approximation of the field, and learning a model of the environmental field. The first component is self explanatory, whereas, the last two components are the key characteristics that define the behavior of all IPP frameworks.

The IPP frameworks generally plan the robot's path based on the data collected. Based on the frequency of this planning, the IPP frameworks can be classified as: non-adaptive algorithms (offline) which commit to a path and do not adjust based on new observations and adaptive algorithms (online) which alter the pre-planned paths on-the-fly based on the new observations. Several non-adaptive algorithms have been suggested in the past to solve near-optimal paths (Meliou et al., 2007; Hollinger and Singh, 2008; Singh et al., 2009; Das et al., 2010) using prior information of the field. However, the prior information for an environmental field may not be available for pre-planning of the robot's path. Such types of applications require the use of adaptive algorithms as the collection of information and path planning have to be synchronized.

The IPP frameworks can also be classified into multi-robot IPP frameworks (Singh et al., 2009; Low et al., 2011, 2012; Kemna et al., 2017) and single-robot frameworks (Zhang and Sukhatme, 2007; Hitz et al., 2017; Mishra et al., 2018). Each of these two classes of frameworks have their own advantages and disadvantages. The planning process for a single-robot framework is less complex compared to a multi-robot frameworks, however, covering large survey areas with a single robot may not be feasible due to limited resources. On the contrary, multi-robot frameworks can easily cover large survey areas by division-of-labor but this division-of-labor adds to the complexity of the multi-robot IPP framework. Moreover, multi-robot frameworks gather more data in a short amount of time and thus require methods that can estimate the field using

large datasets in real-time. Such problems in model learning are currently not addressed and thus limits the usage to small datasets or small survey areas.

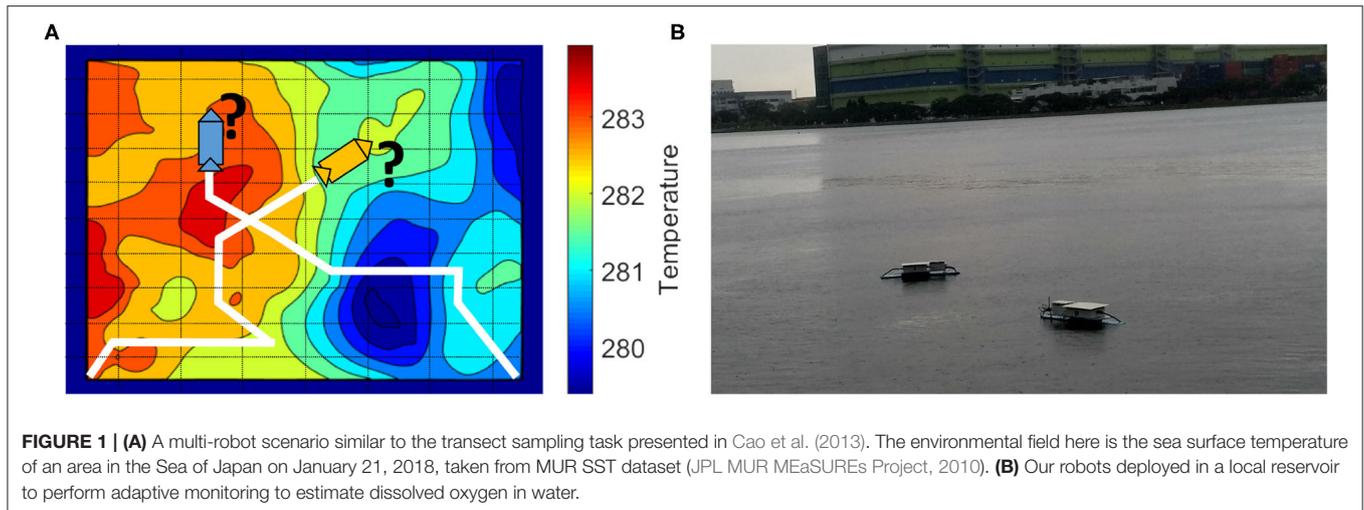
Another challenge in using adaptive algorithms is the online estimation of the survey field, as this estimate governs future waypoints in a robot's path. For example, in the case of water quality monitoring, a good approach may be to use off-the-shelf simulators like Delft3D (Deltares, 2006) or the Regional Oceanic Monitoring System (ROMS) (Moore et al., 2011). However, these simulators generally run on high performance clusters and such computational power is usually not available in robotic platforms. One good approach to combine these simulators with path planning is presented in Smith et al. (2010). In this approach, ROMS uses the data from various sensors to produce velocity profiles on a remote server, which can then be used by the robot for path planning. However, in areas where the sensors for ROMS are not present or the spatial resolution of ROMS's forecast is poor, such an approach will not work.

A commonly used approach in geostatistics (Le and Zidek, 2006; Webster and Oliver, 2007) is to assume that the spatio-temporal environmental field is realized from a probabilistic model called Gaussian processes (GPs). The computational power required for learning a Gaussian process model is comparatively much less than that required by physics-based simulators. Therefore, this approach has been commonly used in path planning (Sukhatme et al., 2007; Zhang and Sukhatme, 2007; Low et al., 2008, 2011, 2012). In Hitz et al. (2017), GPs and an information criterion were used to plan paths for an AUV to segment the environmental field into three different level sets. Similarly, using GPs, a path-planning algorithm based on entropy and information criterion is suggested in Cao et al. (2013). In all of these works, GP regression uses all the data collected during the survey. In a practical scenario, a water-quality sensor (YSI, 2017) can sense data at a frequency of about 1 Hz and thus running a robot with this sensor for an hour will provide about 3,600 data points for learning the model. This means that the data collected during a survey can increase rapidly and therefore, the conventional method of doing GP regression may not be feasible. This problem can be solved by using sparse GPs.

An explanation of how sparse GPs can be integrated into a path planning framework is discussed in Ma et al. (2017). This recent work is directed toward long-term monitoring and overcomes the spatial and temporal changes by updating the GP model based on an information criterion. Although it is a good single-robot framework, the sparse GP point selection can be improved with more data-driven sparse GP variants such as sparse pseudo-inputs Gaussian processes (SPGP) (Snelson and Ghahramani, 2006). The combination of such sparse GP models and time-constrained mission planning for multi-robot frameworks is still lacking.

## 1.2. Practical Constraints in Using a Team of Robots

Our objective is to obtain a good approximation of a scalar environmental field, such as temperature, conductivity, or chlorophyll concentrations, using a team of robots within a



fixed amount of time. We previously published a framework for estimating scalar fields using a single robot (Mishra et al., 2018). A common problem in using a single robot is the limitation on the area it can cover within a finite time, limiting the total collected information. Such problems with single robot scenarios can be easily resolved by using a team of robots to collect more information, however, these robots should be coordinated to collect the information efficiently.

An entropy-based method for multi-robot operation (Cao et al., 2013) generates a set of waypoints using dynamic programming. However, this framework only considers transect environmental fields, where robots can only move along one spatial direction and generate waypoints based on the assumption that fields are anisotropic. Another multi-robot framework uses a lawn mower to obtain preliminary information, and then a leader robot makes decisions to adapt the lawn mower pattern for the team of robots (Chen et al., 2012). Such an approach is helpful for adapting lawn mower patterns, however, following these straight paths consumes time and collects repetitive information. A similar approach is described in Petillo (2015), where the robots maintain a formation and adapt the formation to cover a larger area. Vehicles with motion constraints such as gliders can make use of these frameworks but robots that do not have such strict motion constraints may benefit from a more flexible planning framework.

We are interested in a multi-robot framework that can be used for a team of robots such as AUVs, impose fewer motion constraints, and finish the monitoring task within a fixed amount of time. Moreover, an important component missing in the multi-robot frameworks is the computation time for making decisions. The computation time can be ignored in cases where it is insignificant compared to the overall mission time. However, the framework's task is to finish collecting data within a short amount of time and thus computation time is an important component of our overall mission time. For example, if each decision iteration takes about 5 s to compute and iteration is repeated every 30 s, then during a mission of 600 s, decisions are taken about 20 times. In such a scenario, the computation time

will consume more than 15% of the mission time and thus leave less time for data collection.

### 1.3. IPP Frameworks and Scientific Experiments

Adaptive monitoring frameworks are commonly used for estimating scalar environmental fields such as chlorophyll concentration and temperature. The examples for integrating these estimated fields into biological studies or the relevance for biological studies is still not well-established. Frameworks are designed for scientific experiments such as estimating hotspots or tracking a certain phenomenon, yet the process of using these estimated fields from a biological or geological standpoint is generally missing. This is especially true in the studies to understand the micro-level relationships between the estimated fields and the microorganisms living in it.

Scientists have attempted to establish the connection between the fields estimated using robots and various environmental phenomena. One such work tracks hydrocarbon plumes and bio-degradation at the Deepwater Horizon site (Newman et al., 2008; Camilli et al., 2010). This work focused on developing a framework to observe the bio-degradation of the hydrocarbon plume and it is a good example of tracking a biological phenomenon to understand it at a macro-scale. Another interesting approach for establishing scientific relevance is discussed in Das et al. (2015). In this approach, the authors designed two frameworks, one to make the sampling decisions, and another to estimate the concentration of a pathogen based on the sensor values. The focus of this work was to select samples from a predefined path and estimate the concentration of a particular pathogen.

We are interested in establishing a use case for our framework in identifying the micro-scale species associated with a water-quality parameter. The high concentration regions of these parameters can be both harmful and beneficial to the ecosystem, depending on the biological and chemical characteristics (Darrouzet-Nardi and Bowman, 2011; Zhu et al., 2013; Palta et al., 2014). It is important to find and sample these regions and

discover the associated microbial communities. Sampling from hotspots of oxygen minimum zones has helped in understanding a microorganism's role in terrestrial nitrogen loss in inland waters (Zhu et al., 2015). Therefore, estimating the scalar fields and sampling from the hotspots of parameters such as DO is useful in understanding the environmental processes.

We introduce a multi-robot IPP framework *m-AdaPP* with constraints on mission time for estimating a scalar environmental field. Our aim is to coordinate a team of robots to get a good approximation of the scalar field and finish the overall mission in a fixed amount of time. We make use of a sparse GP method to provide an estimate of the field and the corresponding variance. The paths are evaluated to minimize the overall variance and we include the time taken for this evaluation in our overall mission time. We test the coordination and field estimation performance of our framework using a sea surface temperature dataset in simulation. We also examine the performance of our framework against two multi-robot IPP algorithms, a greedy algorithm and a distributed planning algorithm. We use an approach similar to that shown in Kemna et al. (2017) and the greedy benchmark algorithm as shown in Hitz et al. (2017) to simulate the greedy behavior. The two comparisons with a greedy and distributed algorithm will help us examine the performance gains when using a non-myopic and centralized planning approach. Finally, we compare our framework's performance against the conventional lawn mower patterns for estimating environmental fields, and show that our framework performs well. We also present an approach for integrating our framework into a scientific study.

## 2. SPARSE GAUSSIAN PROCESSES

GP models are commonly used for non-parametric regression problems (Rasmussen and Williams, 2004), such as spatial data modeling (Stein, 2012), image thresholding (Oh and Lindquist, 1999), and soil modeling (Hengl et al., 2004). In a standard GP problem for spatial data regression, the training data set  $\mathbf{D}$  consists of  $N$  vectors each composed of two elements  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$  and corresponding target values  $\mathbf{y} = \{y_n\}_{n=1}^N$  with a Gaussian measurement noise. The likelihood of observed values  $\mathbf{y}$  can be given as  $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$  where  $\mathbf{f}$  is the underlying latent function and  $\sigma^2\mathbf{I}$  is the noise term. Placing a zero mean prior and a covariance function given by  $\mathcal{K}(\mathbf{x}_n, \mathbf{x}_{n'})$  and parameterized by  $\theta$ , the distribution for a new input  $\mathbf{x}$  is given by:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{D}, \theta) = \mathcal{N}(\mathbf{y}|\mathbf{k}_x^\top (\mathbf{K}_N + \sigma^2\mathbf{I})^{-1} \mathbf{y}, K_{x,x} - \mathbf{k}_x^\top (\mathbf{K}_N + \sigma^2\mathbf{I})^{-1} \mathbf{k}_x + \sigma^2), \quad (1)$$

where  $[\mathbf{k}_x]_n = \mathcal{K}(\mathbf{x}_n, \mathbf{x})$ ,  $[\mathbf{K}_N]_{n,n'} = \mathcal{K}(\mathbf{x}_n, \mathbf{x}_{n'})$ , and  $K_{x,x} = \mathcal{K}(\mathbf{x}, \mathbf{x})$ . As it can be observed from (1), the computation time for large datasets will be as high as the prediction, and even the training scales with  $N^3$  due to inversion of the covariance matrix, where  $N$  is the total number of datapoints. Sparse GPs overcome this problem by having sparse approximation of the full GP using only  $M$  points, where  $M \ll N$ . In general, the selection of this subset of  $M$  points is based on information criteria (Seeger et al., 2003).

A common problem with information criterion-based sparse GP methods is the absence of a good method to learn the kernel hyperparameters, because the subset selection and hyperparameter optimization are generally done independently. Moreover, when using *automatic relevance determination* (MacKay, 1998) covariance function, learning bad hyperparameters can adversely affect the prediction performance. The SPGP framework solves this problem by constructing a GP regression model which finds the active subset and hyperparameters in one smooth joint optimization.

### 2.1. Sparse Pseudo-Input Gaussian Processes

In a standard GP model (Rasmussen and Williams, 2004) with zero mean prior, the kernel function is solely responsible for estimating the mapping between the input vector and the target values as shown in (1). Therefore, assuming the hyperparameters of the kernel function are known, the predictive function is effectively parameterized by  $\mathbf{D}$ . In the case of SPGP, this parameterization is done using the pseudo data set  $\bar{\mathbf{D}}$  of size  $M \ll N$ , which has pseudo-inputs  $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_m\}_{m=1}^M$  and corresponding pseudo targets  $\bar{\mathbf{f}} = \{\bar{f}_m\}_{m=1}^M$ . The pseudo targets are denoted as  $\bar{\mathbf{f}}$  instead of  $\bar{\mathbf{y}}$  because these targets do not represent the observed values and therefore, adding the noise variance  $\sigma^2$  can be omitted. The actual prediction distribution has the noise variance and is given as:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{D}, \theta) = \mathcal{N}(\mathbf{y}|\mathbf{k}_x^\top \mathbf{K}_M^{-1} \bar{\mathbf{f}}, K_{x,x} - \mathbf{k}_x^\top \mathbf{K}_M^{-1} \mathbf{k}_x + \sigma^2), \quad (2)$$

where  $[\mathbf{K}_M]_{mm'} = \mathcal{K}(\bar{\mathbf{x}}_m, \bar{\mathbf{x}}_{m'})$  and  $[\mathbf{k}_x]_m = \mathcal{K}(\bar{\mathbf{x}}_m, \mathbf{x})$ , for  $m = 1, 2, \dots, M$ . On comparing (1) and (2), one can clearly observe the reduced computation burden for the inversion of the covariance matrix, from a matrix  $\mathbf{K}_N$  with  $N \times N$  entries to a matrix  $\mathbf{K}_M$  with  $M \times M$  entries. Following the derivation in Snelson and Ghahramani (2006), the predictive distribution is given as a new input  $\mathbf{x}_*$  as:

$$p(\mathbf{y}|\mathbf{x}_*, \mathbf{D}, \bar{\mathbf{X}}) = \mathcal{N}(y_*|\mu_*, \sigma_*^2), \quad (3)$$

where

$$\mu_* = \mathbf{k}_*^\top \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\Omega + \sigma^2 \mathbf{I}) \mathbf{y} \quad (4)$$

$$\sigma_*^2 = K_{*,*} - \mathbf{k}_*^\top (\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}) \mathbf{k}_* + \sigma^2. \quad (5)$$

The derivation of  $\mathbf{Q}_M$  is omitted here for brevity but these are present in Snelson and Ghahramani (2006). The main cost in computing  $\mathbf{Q}_M$  is the inversion of a diagonal matrix (Snelson and Ghahramani, 2006). Using the spatial data as input,  $\mu_*$  will represent the mean predicted field and the variance  $\sigma_*^2$  will constitute the uncertainty in this prediction. The pseudo points  $\bar{\mathbf{X}}$ , parameters  $\theta$ , and noise variance  $\sigma^2$  are learned in one joint optimization given by (6). This joint optimization aims at learning a generative model by maximizing the marginal likelihood with respect to the pseudo points and kernel parameters.

$$p(\mathbf{y}|\mathbf{X}, \theta, \sigma^2) = \mathcal{N}(0, \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{K}_{MN} + \Omega + \sigma^2 \mathbf{I}) \quad (6)$$

**TABLE 1** | Description of all the commonly used notations in all the sections.

| Notation                            | Description   |
|-------------------------------------|---|
| $\tilde{\mathcal{X}}, \mathcal{X}$  | Continuous and discretized spatial domain.  |
| $H$                                 | Total number of robots.   |
| $\mathbf{x}$                        | 2D location vector.   |
| $\mathbf{x}_{t,i}$                  | Location of a robot $i$ at time $t$ .   |
| $\mathbf{x}_{t,1:H}$                | Location of all the $H$ robots at time $t$ .  |
| $\mathbf{x}_t: T_i$                 | Location of robot $i$ from time $t$ to $T$ .  |
| $\mathbf{x}_{t+T_s,1:H}$            | Location of all the $H$ robots at time $t + T_s$ .  |
| $\mathbf{y}_t: T$                   | Measured scalar values by the team of robots between time $t$ and $T$ .                             |
| $\mathcal{Y}(\cdot)$                | Scalar field over the continuous domain $\tilde{\mathcal{X}}$ .                                     |
| $\widehat{\mathbf{D}}_t$            | Data collected from all the $H$ robots till time $t$ .  |
| $\mu_*, \sigma_*^2$                 | Estimated mean and variance by the sparse GP.   |
| $\mathbf{c}_{t,i}$                  | Representative location of the cell at time $t$ , visited by robot $i$ .                            |
| $\mathbf{A}_{\mathbf{c}_{t+T_s,i}}$ | Set of available actions at time $t + T_s$ when a robot $i$ visits a cell $\mathbf{c}_{t+T_s,i}$ .  |
| $V(\mathbf{c}_{t,i})$               | Value function of a cell $\mathbf{c}_{t,i}$ for a robot $i$ at time $t$ .                           |
| $\pi(\mathbf{c}_{t,i})$             | Policy for the cell $\mathbf{c}_{t,i}$ at time $t$ .  |
| $\mathcal{R}(\mathbf{c}_{t,i}, a)$  | Reward for taking action $a$ when the robot $i$ is in cell $\mathbf{c}_{t,i}$ at time $t$ .         |
| $\mathbf{P}_{t,i}$                  | A path for robot $i$ for the remaining time of $T - t$ .  |
| $\mathcal{T}(\mathbf{P}_{t,i})$     | Time for traversing a path $\mathbf{P}_{t,i}$ by robot $i$ at time $t$ .                            |
| $\mathbf{P}_{t,i}^0$                | Starting location of the path for the robot $i$ at time $t$ .                                       |
| $\Lambda_{t,i}$                     | A set of all the paths $\mathbf{P}_{t,i}$ for a robot $i$ for the remaining time $T - t$ .          |
| $\widehat{\mathbf{P}}_t$            | A set of paths for all the $H$ robots for the remaining time $T - t$ .                              |
| $\widehat{\Lambda}_t$               | A set containing all the paths for all the robots for the remaining time $T - t$ .                  |
| $T_s$                               | Time interval for planning iterations.  |
| $\Phi_{t+T_s}$                      | Set of all available actions for the team of robot at time $t + T_s$ .                              |
| $\varphi'_{t+T_s}$                  | One set of actions for the team of robots at time $t + T_s$ .                                       |
| $\mathcal{U}(\varphi'_{t+T_s})$     | Sum of cell variances reached by all the robots on taking the actions given by $\varphi'_{t+T_s}$ . |
| $\vartheta_{T-(t+T_s)}(\cdot)$      | Represents the potential of reducing variance in the remaining time $T - (t + T_s)$ .               |

We follow the suggestions given in Snelson and Ghahramani (2006) for initialization of  $M$  pseudo points and the kernel function and learn these parameters to get a representative model of the collected data. Moreover, the scalar environmental fields can be non stationary (Cao et al., 2013) and up to a certain extent, SPGP is capable of modeling non-stationary GP processes through its pseudo-inputs, which gives it an edge over other sparse GP methods.

### 3. PROBLEM FORMULATION

We follow the common notations stated in **Table 1** throughout our formulation and the suggested solution for consistency. Broadly, our problem statement is to find a path for a team of  $H$  robots and collect representative data to provide a good estimate of the environmental field and finish this task within a fixed amount of time  $T$ . This statement can be represented as:

$$\arg \min_{\widehat{\mathbf{P}}_t \in \widehat{\Lambda}_t} \frac{1}{|\mathcal{X}|} \int_{\tilde{\mathcal{X}}} (\mathcal{Y}(\mathbf{x}) - \widehat{\mathcal{Y}}(\mathbf{x}, \widehat{\mathbf{D}}_t \cup \widehat{\mathbf{D}}_{T-t, \widehat{\mathbf{P}}_t}))^2 d\mathbf{x}, \quad (7)$$

such that,

$$\mathcal{T}(\widehat{\mathbf{P}}_t) \leq T - t, \text{ and} \quad (8)$$

$$\widehat{\mathbf{P}}_t^0 = \mathbf{x}_{t,1:H}, \quad (9)$$

where  $\widehat{\mathbf{P}}_t$  is a set containing one path for each robot and given as:

$$\widehat{\mathbf{P}}_t = \{\mathbf{P}_{t,1}, \mathbf{P}_{t,2}, \mathbf{P}_{t,3}, \dots, \mathbf{P}_{t,H}\}, \quad (10)$$

and each of these paths  $\{\mathbf{P}_{t,1}, \mathbf{P}_{t,2}, \mathbf{P}_{t,3}, \dots, \mathbf{P}_{t,H}\}$  is a set of locations given by  $\{\mathbf{x}_{t:T,1}, \mathbf{x}_{t:T,2}, \mathbf{x}_{t:T,3}, \dots, \mathbf{x}_{t:T,H}\}$ . Similarly,  $\widehat{\Lambda}_t$  is a set containing all the paths for each robot and it is given as:

$$\widehat{\Lambda}_t = \{\Lambda_{t,1}, \Lambda_{t,2}, \Lambda_{t,3}, \dots, \Lambda_{t,H}\}. \quad (11)$$

The function  $\mathcal{Y}(\cdot)$  in (7) is the field over the spatial domain  $\tilde{\mathcal{X}}$  and  $\widehat{\mathbf{D}}_t$  is the data collected by all the robots and thus  $\widehat{\mathbf{D}}_t = \{\mathbf{D}_{t,1}, \mathbf{D}_{t,2}, \mathbf{D}_{t,3}, \dots, \mathbf{D}_{t,H}\}$ , where  $\mathbf{D}_{t,i}$  is the data collected by robot  $i$  until time  $t$ . The function  $\widehat{\mathcal{Y}}(\cdot, \cdot)$  is the estimated function of the field at time  $t$  using the collected data  $\widehat{\mathbf{D}}_t$  and the data yet to be collected  $\widehat{\mathbf{D}}_{T-t, \widehat{\mathbf{P}}_t}$  by traversing paths given by  $\widehat{\mathbf{P}}_t$ . The path  $\mathbf{P}_{t,i}$  and the set of collection of paths  $\Lambda_{t,i}$  in (10) and (11) represent the candidate paths for robot  $i$ . Moreover,  $\widehat{\mathbf{P}}_t^0$  in (9) are the starting locations, which are also the locations of all the  $H$  robots at time  $t$ . All the paths in the set  $\widehat{\Lambda}_t$  start from the locations given by  $\widehat{\mathbf{P}}_t^0$ . Finally, the function  $\mathcal{T}(\cdot)$  provides an estimate of the time to traverse a path. In our problem statement, we have defined the measure of goodness as a low mean squared error over the complete spatial domain. The current form of the problem statement is not solvable as we cannot get the information about  $\mathcal{Y}(\cdot)$  without sampling or visiting locations and thus without actually traversing a set of paths  $\widehat{\mathbf{P}}_t$ , we cannot obtain the target values  $\mathbf{y}_t: T = \{y_i\}_{i=t}^T$  for yet to be visited locations. To overcome this, we can make use of characteristics of a GP model to make problem (9) solvable. The function  $\widehat{\mathcal{Y}}(\cdot, \cdot)$  can be learned using a GP model and it can be written as  $\mathcal{N}(\mu_*, \sigma_*^2)$ , where  $\mu_*$  should represent a close approximation of  $\mathcal{Y}(\cdot)$  if the learned GP model is a good fit and the overall variance  $\sigma_*^2$  is low. Therefore, we can re-write (7) as:

$$\arg \min_{\widehat{\mathbf{P}}_t \in \widehat{\Lambda}_t} \frac{1}{|\mathcal{X}|} \sum_{\mathcal{X}} \sigma_*^2(\mathbf{x}, \widehat{\mathbf{D}}_t, \widehat{\mathbf{P}}_t). \quad (12)$$

It is important to take note of two changes between (7) and (12). First, we have replaced  $\widehat{\mathbf{D}}_{T-t, \widehat{\mathbf{P}}_t}$  with just  $\widehat{\mathbf{P}}_t$  as we can get an estimate of the variance without sensing the target values and only the spatial locations given by  $\widehat{\mathbf{P}}_t$  are sufficient. However, the estimated variance depends on  $\widehat{\mathbf{D}}_{T-t}$  and it will be updated using (5) whenever the robot collects more data  $\widehat{\mathbf{D}}_t$ . Therefore, our planning problem can be seen as collecting good data such that the overall variance becomes low.

Second, the problem statement given by (7) is in the continuous domain  $\tilde{\mathcal{X}}$ . This means the number of paths in

the set  $\widehat{\Lambda}_t$  will be large and searching for the optimal path  $\widehat{\mathbf{P}}_t^*$  that satisfies our problem statement will be difficult. A common approach to reduce such complexity is to discretize the continuous domain  $\mathcal{X}$  into a grid  $\mathcal{X}$ . In this scenario, each location  $\mathbf{x}$  will generally have eight neighbors and thus for each location the decision will be to select which of these neighbors to visit. Finally, the constraints on (12) will be:

$$\mathcal{T}(\mathbf{P}_{t,i}) + \tau \leq T - t, \text{ and} \quad (13)$$

$$\mathbf{P}_t^0 = \mathbf{x}_{t,i} \forall i \in [1, H], \quad (14)$$

where the new addition  $\tau$  in comparison to (8) represents the computation time for each decision and  $\mathbf{x}_{t,i}$  represents the location of robot  $i$  at time  $t$ . The constraints given by (13) represent that each robot will have less than  $T - t$  time available for collecting data. However, we can absorb  $\tau$  inside  $\mathcal{T}(\mathbf{P}_{t,i})$  if the computation can be done while traversing. This will require taking a decision for the next location while collecting data. The current formulation given by (12) will not allow this as the decision made at time  $t$  is possible only after collecting all the data  $\widehat{\mathbf{D}}_t$  until time  $t$ . However, we can use the data  $\widehat{\mathbf{D}}_t$  to make a decision for the next location  $\mathbf{x}_{t+1}$  and collect more data while traveling from  $\mathbf{x}_t$  to  $\mathbf{x}_{t+1}$ . This will change the problem statement to:

$$\arg \min_{\widehat{\mathbf{P}}_t \in \widehat{\Lambda}_{t+1}} \frac{1}{|\mathcal{X}|} \sum_{\mathcal{X}} \sigma_*^2(\mathbf{x}, \widehat{\mathbf{D}}_t, \widehat{\mathbf{P}}_t), \quad (15)$$

such that,

$$\tau \leq \mathcal{T}(\{\mathbf{x}_{t,i}, \mathbf{x}_{t+1,i}\}) \text{ and} \quad (16)$$

$$\mathcal{T}(\mathbf{P}_{t,i}) \leq T - t \text{ and} \quad (17)$$

$$\mathbf{P}_{t,i}^0 = \mathbf{x}_{t+1,i} \forall i \in [1, H], \quad (18)$$

where  $\widehat{\Lambda}_{t+1}$  represents the set of all paths for each robot  $i$  from its next location  $\mathbf{x}_{t+1,i}$ . This formulation changes (13) to (17) but introduces a new constraint given by (16), which suggests that computation time should be less than or equal to the time taken by the robots to travel to the next location. The set  $\widehat{\Lambda}_{t+1}$  can be visualized as a state space too. This state space will be a convolution of multiple state spaces given by  $\{\Lambda_{t+1,i}\}$  and its starting state given as  $\mathbf{s}_t = \{\mathbf{x}_{t+1,i}\} \forall i \in [1, H]$ . The state space of the set  $\widehat{\Lambda}_{t+1}$  at each planning iteration aims to reduce the variance and this is similar to the problem of selecting locations as described in Singh et al. (2009), which is shown to be an NP-hard search problem. Therefore, we need a framework to transform this search problem that can be solved in polynomial time and provide a good estimate of the environmental field.

## 4. MULTI-ROBOT PLANNING FRAMEWORK

We suggest a centralized framework, named as *m-AdaPP*, to efficiently search through the state space given by  $\widehat{\Lambda}_{t+1}$  and collect data using the kernel information to get a good estimate

of our field. This algorithm follows the basic IPP framework and thus has the three components, which are planning, model learning, and collecting data. As discussed in the section before, we learn the model and plan for the next location while the robots are traveling and collecting more data. We make use of the spatial decomposition approach as explained in Mishra et al. (2018) and reduce our search space by discretizing the grid into cells.

There are three constraints on our planning as shown in our problem formulation. These are the limits on each robot's total mission time  $T$ , bounds on the computation time used for planning, and each robot's starting point. Although the planning is done over cells instead of locations, this does not mean that robots do not collect data while traveling from one cell to another. The data are collected as and when the sensors provide a scalar value of the field, defined by the sensor's frequency rate. These data are then stamped against the current location of the robot and sent to a central server. This server uses the data for estimating the environmental field.

In the discretized area, the representative location of each cell changes based on the variance in that cell. This results in each robot traversing different lengths of paths, which means robots reach their next waypoint at different times. Therefore, the update of the collected data is asynchronous and planning decisions are made using partial information. We bring synchronization between the team of robots by dividing the total time  $T$  into intervals of  $T_s$ , where by the end of each interval the robots reach their waypoints. Therefore, this time interval  $T_s$  is sufficient for a robot to reach the neighboring cell, even when traversing at the average speed. The addition of this synchronization time step also transforms the decision step from  $t + 1$  to  $t + T_s$  and thus the framework uses this time interval to plan for the paths ahead of the next synchronization event.

### 4.1. Multi-Robot Path Planning With No Constraints

We make use of *single-robot dynamic programming (DP)* along with the spatial decomposition algorithm discussed in Mishra et al. (2018) to explain our multi-robot path planning algorithm. This *single-robot DP* algorithm can be defined as a Markov decision process (MDP). The formulation as a MDP will require states to be defined by the cells, actions as the moves available in each of these cells, the transition probabilities as 1, and the reward function given by  $R(\cdot, \cdot)$ . Extending the single-robot algorithm to a multi-robot scenario requires two modifications. First, the robots should be coordinated to explore an area in a collaborative manner. This is similar to the problem solved in Singh et al. (2009) using a sequential decision algorithm. This algorithm decides a path for one robot first, which is followed by path allocation to the second robot and then sequentially to the remaining robots. However, we are concerned only with the next waypoint in the case of planning with no constraints. The second necessary modification is to prevent the collision between two robots, which can be achieved by having negative rewards for each robot's current location. For the ease of notation, we denote the representative location of a cell visited by a robot  $i$  at time  $t$  as  $\mathbf{c}_{t,i}$ . As we are interested in planning for one-time ahead, the

update rules for multi-robot case can be given as:

$$V(\mathbf{c}_{t,i}) = \max_{a \in \mathbf{A}_{\mathbf{c}_{t,i}}} [\mathcal{R}(\mathbf{c}_{t,i}, a) + \gamma V(\mathbf{c}_{t+T_s,i})], \quad (19)$$

$$\pi(\mathbf{c}_{t,i}) = \arg \max_{a \in \mathbf{A}_{\mathbf{c}_{t,i}}} V(\mathbf{c}_{t,i}), \quad (20)$$

where

$$\mathcal{R}(\mathbf{c}_{t,i}, a) = \begin{cases} \frac{\sigma_{*,c_t}}{\|\mathbf{c}_{t,i} - \mathbf{c}_{t+T_s,i}\|} & \text{if } \mathbf{c}_{t+T_s,i} \notin \mathbf{x}_{t+T_s,1:H} - \{\mathbf{x}_{t+T_s,i}\} \\ -\epsilon & \text{if } \mathbf{c}_{t+T_s,i} \in \mathbf{x}_{t+T_s,1:H} - \{\mathbf{x}_{t+T_s,i}\} \end{cases},$$

$\mathbf{x}_{t,1:H}$  represents the current location of all the robots,  $\mathbf{c}_{t+T_s,i}$  represents the location of the cell that the robot  $i$  will reach at time  $t + T_s$ , and  $\epsilon$  is the value of the negative reward. We run one full cycle of policy iteration using DP for robot 1 and obtain the optimal policy given by  $\pi^*(\cdot)$ . Using this policy, we get the future location of robot 1, given by  $\mathbf{c}_{t+T_s,1} = \pi^*(\mathbf{c}_{t,i})$  and thus we update  $\mathbf{x}_{t+T_s,1} = \mathbf{c}_{t+T_s,1}$ . We also update this new location for robot 1 in the location set of all robots  $\mathbf{x}_{t+T_s,1:H}$ . This update of the location in the set of locations  $\mathbf{x}_{t+T_s,1:H}$  makes sure that robot 2 and the remaining robots do not visit the same cell where robot 1 will be at the next time step. We run such cycles sequentially for all the  $H$  robots and obtain the next respective waypoints. We name this algorithm the *multi-robot DP*.

## 4.2. Multi-Robot Path Planning With Temporal Constraints

Introducing time constraints to the multi-robot framework explained above is not straightforward. The new waypoints generated using the above framework may not be optimal given the temporal constraints  $T - t$ . Therefore, we need to find a combination of actions for different robots that would reduce the overall variance within the remaining time. Let  $\Phi_{t+T_s}$  denote this combinatorial set of all actions  $\mathbf{A}_{\mathbf{c}_{t+T_s,i}} \forall i \in [1, H]$  at time  $t + T_s$ . We define another combinatorial set  $\varphi_{t+T_s}$ , which is a subset of  $\Phi_{t+T_s}$  representing one action for each robot. From the set  $\Phi_{t+T_s}$ , we remove the states where the next action for two or more robots will result in a collision. Therefore, the optimal combination of action  $\varphi_{t+T_s}^*$  at time  $t + T_s$  can be given by:

$$\arg \max_{\varphi_{t+T_s}^* \in \Phi_{t+T_s}} \mathcal{U}(\varphi_{t+T_s}^*) + \eta \vartheta_{T-(t+T_s)}(\varphi_{t+T_s}^*), \quad (21)$$

where  $\mathcal{U}(\varphi_{t+T_s}^*)$  is a function that gives the sum of variances of cells that will be visited due to the combination of actions in  $\varphi_{t+T_s}^*$ ,  $\eta$  is a discounting factor, and  $\vartheta_{T-(t+T_s)}(\varphi_{t+T_s}^*)$  represents the potential of reducing variance within the remaining time  $T - (t + T_s)$  by taking the combination of actions given by the set  $\varphi_{t+T_s}^*$ .

Interestingly, calculating the variance has no direct dependency on the target values  $\mathbf{y}_{t:T}$  as shown in (5). This suggests that once the kernel function is learned using the collected data, we can estimate the change in variance over the field. We use this characteristic to get an estimate of  $\vartheta_{T-(t+T_s)}(\varphi_{t+T_s}^*)$ . The variance after taking a path can be

---

### Algorithm 1: *m-AdaPP* - multi-robot adaptive path planning

---

**Data:** Starting points ( $\mathbf{x}_{t,1:H}$ ), total mission time ( $T - t - T_s$ ), SPGP parameters ( $M$ )

**Result:** Estimate field  $\mu_*$

```

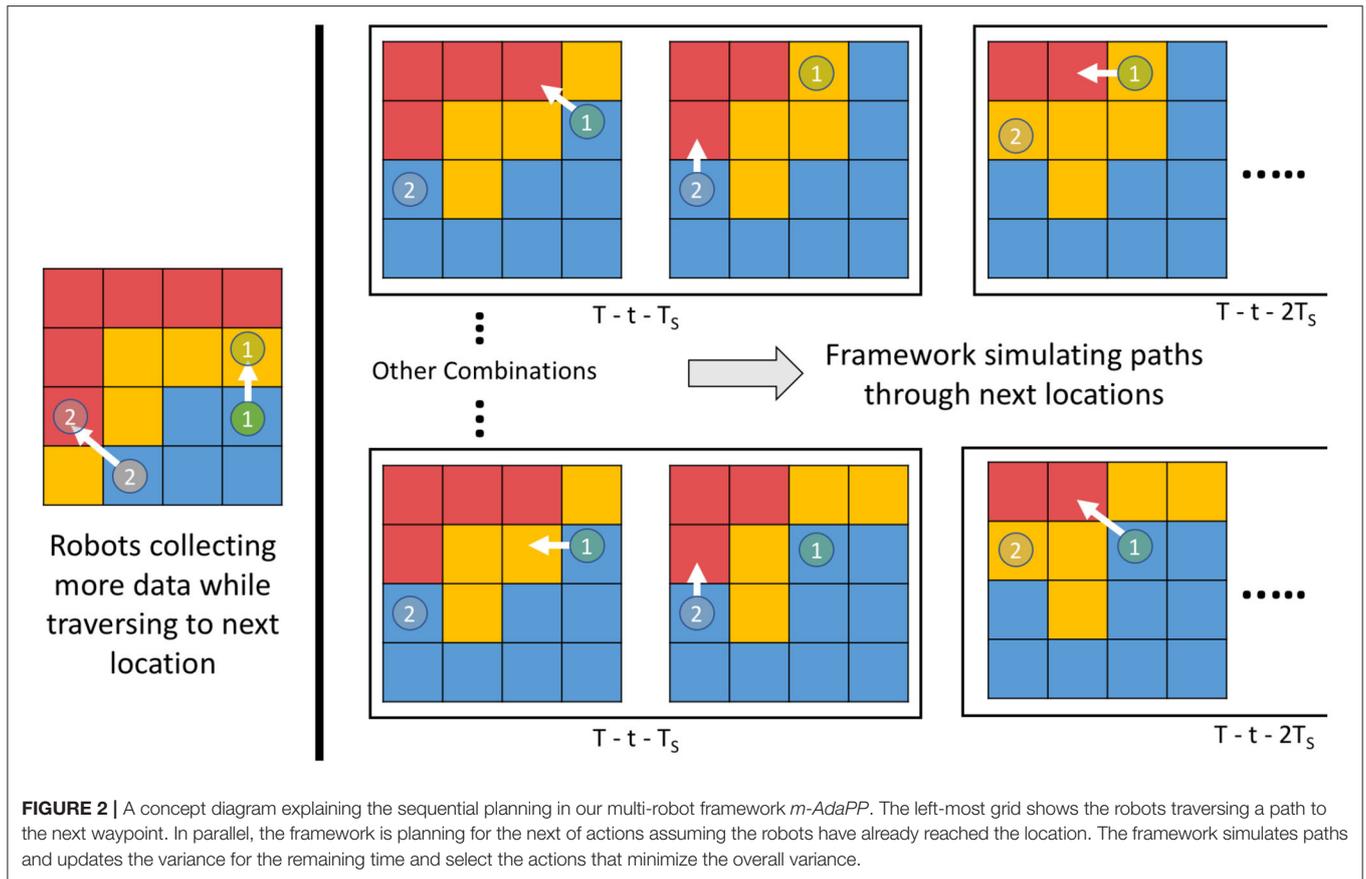
/* Initialization */
1 Run sampling decomposition using resolution  $G$  and set  $\sigma_*^2$  as constant;
/* Algorithm loop */
2 while  $t < T$  do
3   Construct the set  $\Phi_{t+T_s}$ ;
4   foreach  $\varphi_{t+T_s}^* \in \Phi_{t+T_s}$  do in parallel
5     Estimate  $\vartheta_{T-(t+T_s)}(\varphi_{t+T_s}^*)$  by simulating planning using multi-robot DP and SPGP kernel;
6     Calculate and store  $\mathcal{U}(\varphi_{t+T_s}^*) + \eta \vartheta_{T-(t+T_s)}(\varphi_{t+T_s}^*)$ ;
7   Find the set  $\varphi_{t+T_s}^*$  that solves (21);
8   Wait for time interval given by  $T_s$  to be over;
9   Take the actions given in  $\varphi_{t+T_s}^*$ ;
10  Update the recently collected data training data  $\widehat{\mathbf{D}}_{t+T_s}$  and the time  $t = t + T_s$ ;
11  Initialize  $\theta$  using full-GP on a random subset of collected data  $[\widehat{\mathbf{D}}_t(\text{rand}(M))]$ ;
12  Initialize  $M$  pseudo points  $\bar{\mathbf{x}}$  with a random subset of visited locations  $\mathbf{x}_{1:t,1:H}(\text{rand}(M))$ ;
13  Learn SPGP model using  $[\widehat{\mathbf{D}}_t, \theta, \bar{\mathbf{x}}]$  and obtain  $\mu_*$  and  $\sigma_*^2$  over the field  $\mathcal{X}$ ;
14  Run sampling decomposition using resolution  $G$  and  $\sigma_*^2$ ;

```

---

estimated using (5) and the remaining overall variance in the field will give the estimate for  $\vartheta_{T-(t+T_s)}(\varphi_{t+T_s}^*)$ . However, obtaining just this value will not solve our problem. We need to coordinate a team of robots and select the best available option given the remaining time  $T - (t + T_s)$ . We still need to search through the set  $\Phi_{t+T_s}$  to select a set of actions at time  $t + T_s$ .

We solve the problem of coordination between the robots by using *multi-robot DP* at each simulated planning iteration and provide a combination of actions. We do this in two steps. First, we run one full iteration of *multi-robot DP* and obtain a set of actions  $\varphi_{t+T_s}^*$ . Second, we reduce the total time by  $T_s$  and update the variance of the cells based on the paths the robots will take due to the actions given by  $\varphi_{t+T_s}^*$ . We re-run the *multi-robot DP* algorithm to find the next set of actions  $\varphi_{t+2T_s}^*$  using the updated variance. We iterate over these two steps until the mission time is over  $t = T$ . Using this approach, we get an estimate of  $\vartheta_{T-(t+T_s)}(\cdot)$  and thus we can evaluate the value of the combination  $\varphi_{t+T_s}^*$  given by (21). Similarly, we can use this to find the values for all the combinations given by the set  $\Phi_{t+T_s}$ . Once we have the values for all the actions, we can use (21) to find the set of actions for the robots for time  $t + T_s$ . An example of these steps is illustrated as a diagram in **Figure 2**. All these steps are repeated whenever the training dataset  $\widehat{\mathbf{D}}_t$  is updated, which will be at a regular interval of  $T_s$  and thus bring the adaptive nature to



the *m-AdaPP* framework. Our overall framework is presented in Algorithm 1 and a graphical illustration of it is shown in Figure 3.

There are two important points to note about our framework. First, the decisions are made sequentially but it does not mean the robots also move sequentially. Once a decision is made, all the robots move to their next location simultaneously within time  $T_s$ . Second, the calculation of  $\vartheta_{T-(t+T_s)}(\cdot)$  for one set of action in  $\Phi_{t+T_s}$  is independent of the other set of action. This provides an opportunity to estimate the value for  $\vartheta_{T-(t+T_s)}(\cdot)$  for all the sets of actions in parallel. This helps in reducing the overall computation time of our framework.

### 4.3. Field Prediction Using SPGP

We make use of the same kernel function used in single-robot frameworks. It is defined by  $K(\cdot, \cdot)$ :

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = \alpha \exp\left(\frac{1}{2} \sum_{l=1}^2 b_l (x_{n,l} - x_{n',l})^2\right), \quad (22)$$

where  $\alpha$ ,  $b_1$ , and  $b_2$  are the parameters of the kernel function,  $\mathbf{x}_n$  and  $\mathbf{x}_{n'}$  represent two different locations, and  $x_{n,l}$  represents the value for the  $l$  dimension of  $\mathbf{x}_n$ . After adding the Gaussian noise model, the hyperparameters of the sparse GP are given by  $\theta = \{\alpha, b_1, b_2, \sigma^2\}$  and pseudo inputs  $\bar{\mathbf{x}}$ . Following the suggestions given in Snelson and Ghahramani (2006), we initialize the pseudo points with random spatial locations from the collected data and initialize the kernel hyperparameters by learning a full-GP model

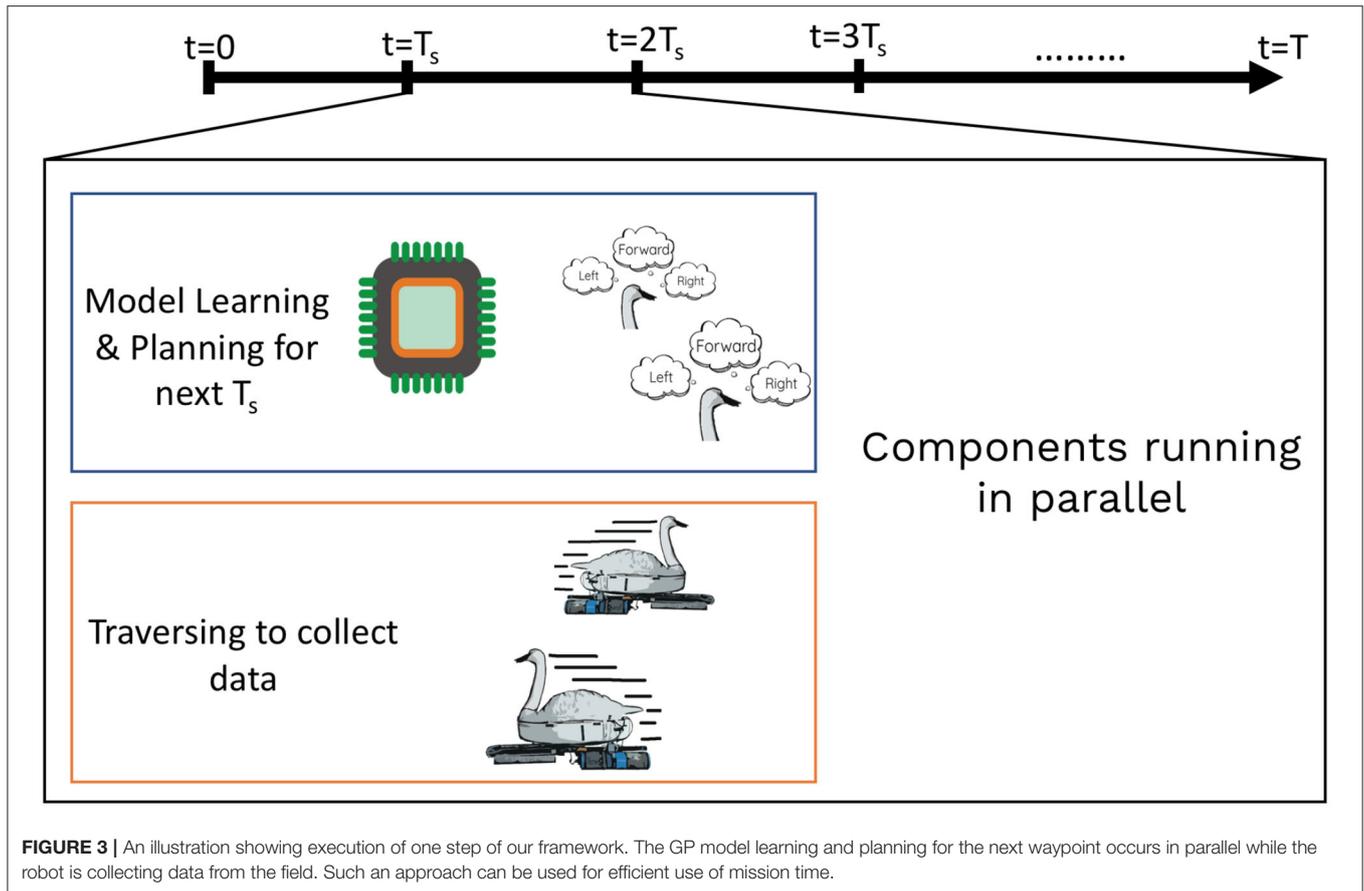
with the same kernel function but using only a small subset of the dataset.

## 5. EXPERIMENTAL RESULTS

We performed two sets of experiments to test the performance of our framework. We first examined the coordination within the team of robots and later we did experiments in Singapore waters to compare the fields estimated by our framework and lawn mower patterns. Finally, we examined the biological relevance of the fields estimated using our framework.

### 5.1. Simulations to Test the Coordination Efficiency

In our previous work (Mishra et al., 2018; Mishra, 2019), we have shown via simulations that our single robot adaptive algorithm performs better as compared to lawn mower and other commonly used search techniques. The objective of these simulations was primarily to establish that our framework is capable of coordinating a team of robots and provide a good estimate of the field. We used field data of sea surface temperature (SST) provided by the Jet Propulsion Laboratory (JPL MUR MEaSUREs Project, 2010). We extracted data for two regions of  $200 \times 200 \text{ km}^2$  each, and mapped each to a field with an area of  $200 \times 200 \text{ m}^2$ . This scaling was done to retain essential features of a scalar temperature field, but also to include an area which can be explored within a practical value of mission time  $T$ . The main



feature of this field is its scalar nature and not that it represents the sea surface temperature. It can be easily compared to the fields of vegetation spread, air quality, or ash plumes. We denote the two scaled temperature fields shown in **Figure 4A** as Field 1 and **Figure 5A** as Field 2 for the following discussions.

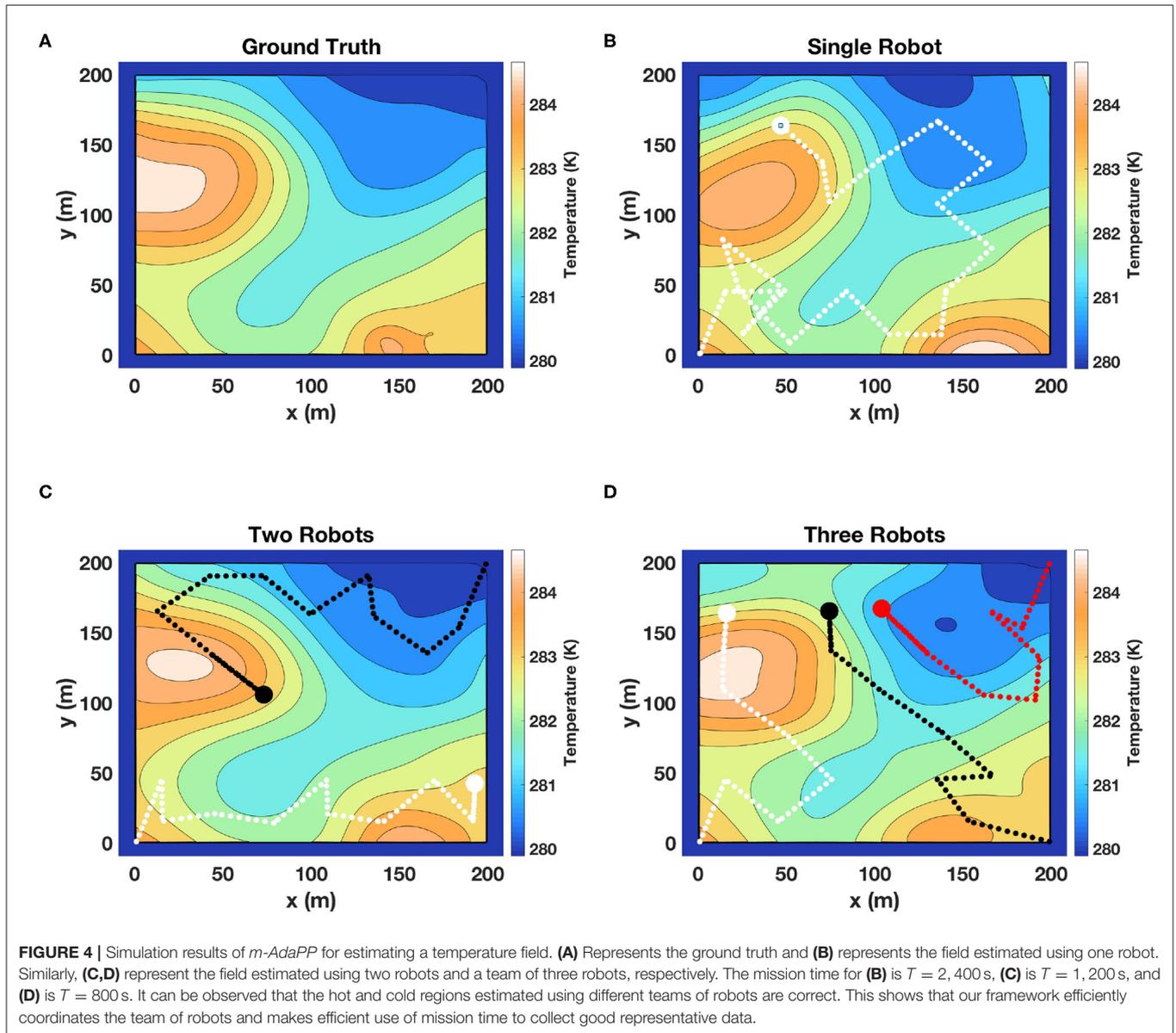
The maximum speed of the robots used in environmental monitoring is generally low. This is to make sure that the robots do not cause substantial disturbance to the environment they is sensing. For example, the maximum speed of our water-quality sensing robot, NUSwan (Koay et al., 2017), is 1 m/s. However, the average speed of NUSwan with external disturbances such as strong winds or waves is about 0.3 m/s. We use this speed to define the value of  $T_{cleveres}$ . Following the grid size  $G = 30$  m, the average time required for traveling from one cell to another cell will be at least 100 s. Therefore, we set the value of  $T_s$  as 120 s giving the vehicle sufficient time to reach the next cell.

We learned the SPGP model with  $M = 50$  pseudo data points. Similar to the single robot framework, we initialized the pseudo points with  $M$  random points of the total dataset and ran a full GP regression to initialize the hyperparameters of our kernel function. The simulation experiments were implemented in MATLAB. For SPGP, we took the MATLAB code provided by the authors (Snelson and Ghahramani, 2006) and modified it for spatial regression application. The simulations were run on a hexa-core Intel i7 processor with 32 GB of RAM.

We simulated teams consisting of a maximum of three robots. We examined the coordination within the team of robots by providing less mission time for the teams with a higher number of robots. This means that the team with two robots has less time compared to a single robot. If the framework is able to coordinate the team of two robots well, the performance of these two simulations should be comparable. For our simulation setup, we set the mission time  $T$  as 2,400 s for a single robot, 1,200 s for a team of two robots, and 800 s for a team of three robots.

Note that the mission time  $T$  for a single robot here is 2,400 s, which is much higher than the mission times set in our previous work (Mishra et al., 2018). This difference is due to the assumed vehicle speed, and a relationship can be seen in terms of distance traveled: a vehicle with speed 0.3 m/s travels around 700 m in 2,400 s. Whereas, the same vehicle with an increase of 1 m/s in speed travels the same distance in 700 s. Therefore, our limit on mission time in the current setup is not substantially different from the setup in Mishra et al. (2018). Moreover, our average computation time for the team of three robots after parallelization was about 23 s, which is much less than  $T_s$  and thus satisfies the constraint on  $\tau$  given by (16).

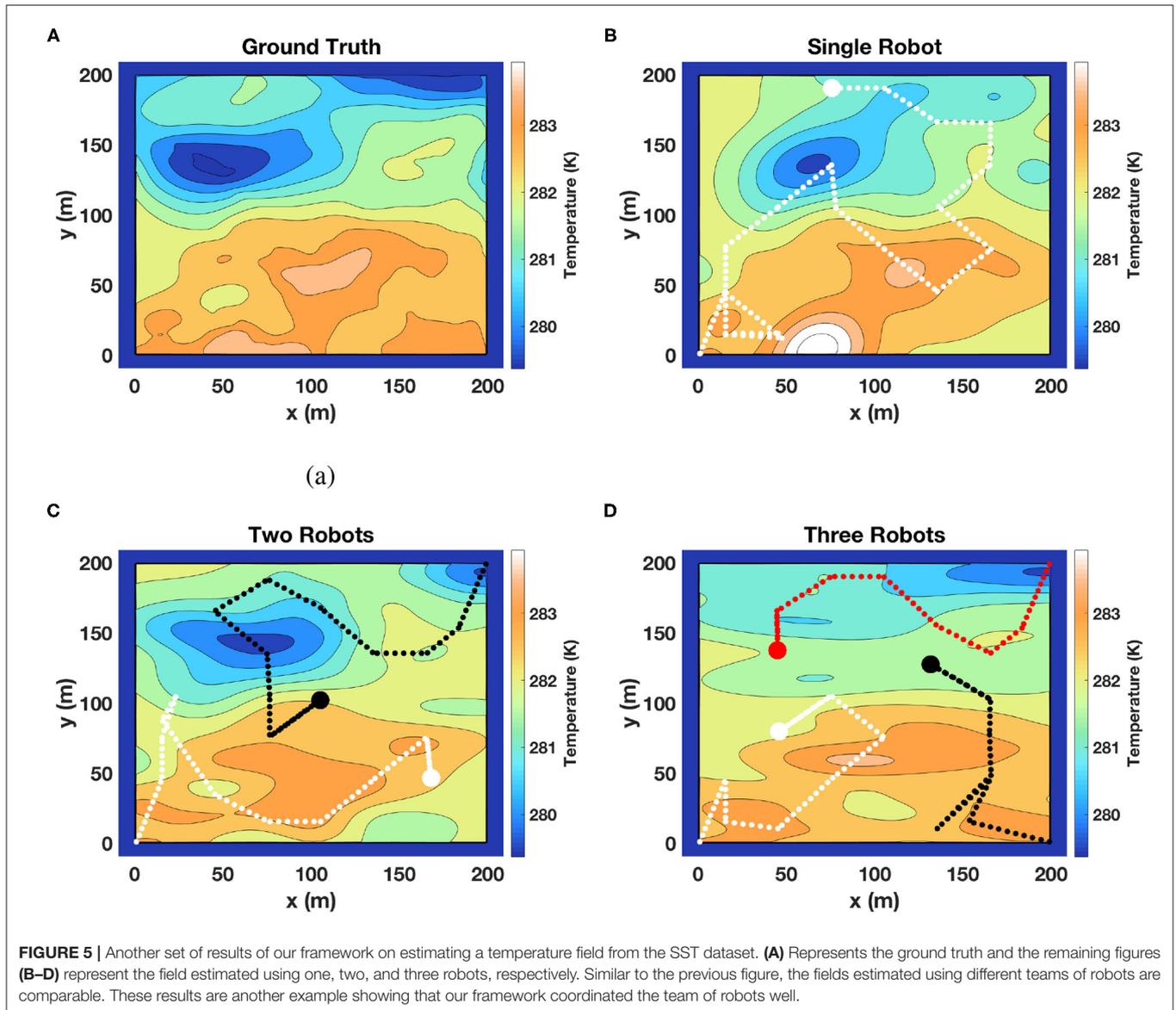
The results of the fields estimated using *m-AdaPP* are shown for one simulation run in **Figures 4, 5**. It is clear from the figures that the estimated hot and cold regions in



our framework are correct and the overall estimated fields are similar for teams with different numbers of robots. We also calculated the mean absolute error (MAE) over all the locations in the entire field and used it as a measure of performance in estimating the fields. We use this metric to examine the coordination efficiency of our framework. The MAE results for one simulation run are presented in Figure 6.

It can be observed from Figure 6 that our framework's performance is similar for different teams of robots. The mission time for each team of robots is proportional to the number of robots in each team. This means that the amount of data collected by a single robot in  $T = 2,400$  s will be similar to the amount of data collected by a team of two robots in  $T = 1,200$  s. A similar performance between

these two setups will show that our framework is able to efficiently coordinate the team of robots. Therefore, the similar MAE values in Figure 6 for different teams of robots and for different fields is a good indication that our framework is capable of coordinating the team efficiently. It can be also observed from Figure 6 that the performance of multi-robot teams is less monotonic. This could be due to the random initialization of SPGP and thus we also repeated the simulations over 10 runs for each team of robots and recorded the MAE. The main difference between these 10 runs was the random initialization of the SPGP model and the corresponding planning using this SPGP model. These results are presented in Figure 7, and it can be observed that our framework shows a consistent monotonic performance over multiple runs. The results in Figure 7 give an overview of the performance over



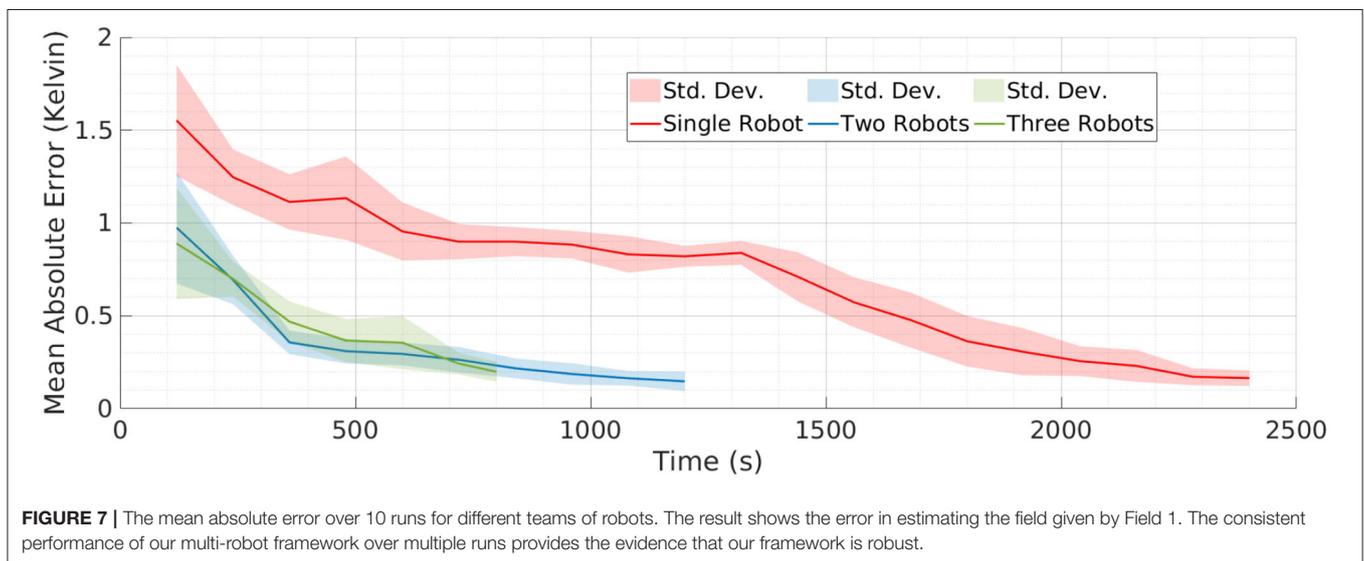
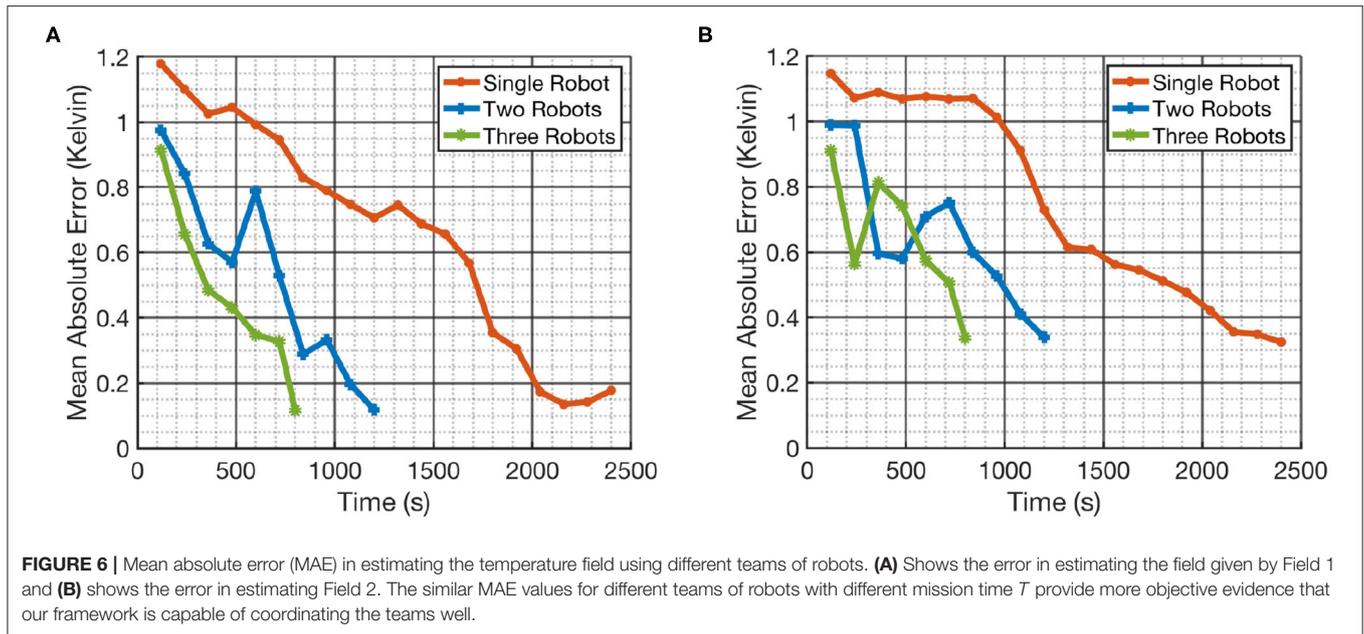
10 runs whereas **Figures 4–6** are results from a randomly selected instance.

## 5.2. Performance Comparison With Greedy IPP in Simulations

Our framework searches for a combination of actions for the team of robots that satisfies (21). This equation includes both the short term goals, denoted by  $\mathcal{U}(\cdot)$ , and the long term goals, denoted by  $\vartheta_{T-(t+T_s)}(\cdot)$ . Interestingly, removing the term  $\vartheta_{T-(t+T_s)}(\cdot)$  from (21) will shift the framework's focus to the sum of variance of neighboring cells and thus convert our framework into a greedy IPP. Moreover, removing this term will also relax the dependence on future moves and thus simulate a myopic planning approach. The time bounds will only be present to stop the simulation and not constrain the framework's planning or model learning.

The key difference between the greedy IPP and our framework is the selection of actions at any given time  $t$ . Both the frameworks use the same sparse GP method and the actions are selected in a centralized manner. The performance of this greedy framework can be thus used as a benchmark and effectively compare two different IPP approaches, myopic and non-myopic.

We simulated the greedy IPP using the simulation setup explained in the previous section. The greedy IPP and *m-AdaPP* are both given the same amount of time for a team of two robots and we simulated 10 runs for both the fields. We calculated the MAE values for all the runs and the end results are shown in **Figure 8**. It can be clearly observed that *AdaPP* performs better when compared to a greedy IPP. These results are encouraging as it shows that our non-myopic planning approach performs well and efficiently coordinates the team of robots within the given time.

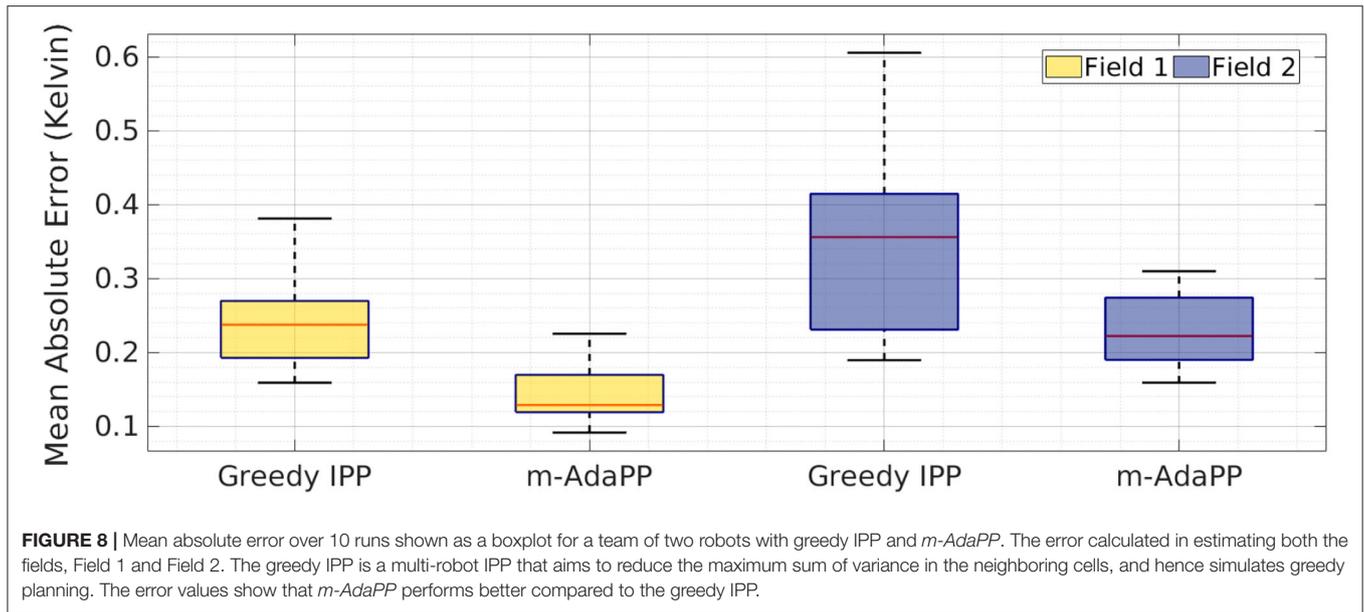


### 5.3. Simulations to Compare Performance With a Distributed Implementation of *m-AdaPP*

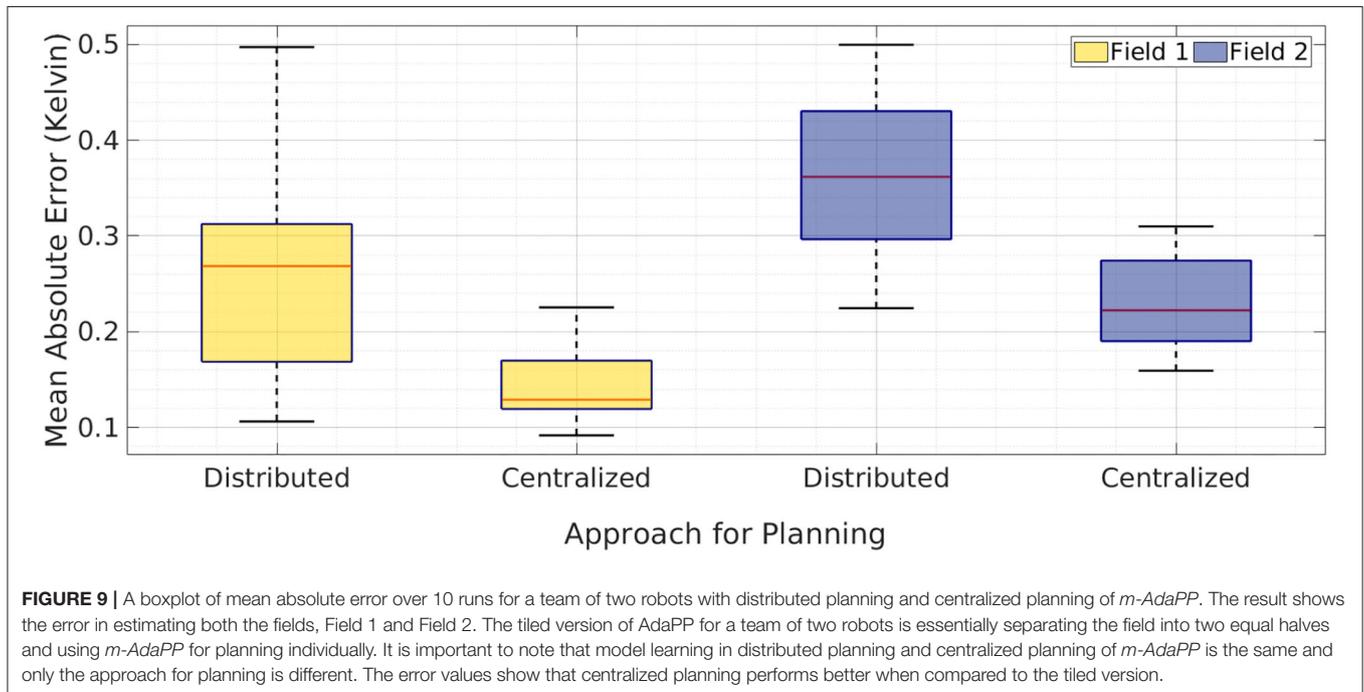
Our framework makes use of centralized planning for coordinating the team of robots and this centralized planning can be distributed for the team of robots using different approaches. One of these approaches can be splitting the area into proportional areas to the number of robots and perform planning for each robot in its respective area. It is important to note that in this approach only the planning will be performed separately for each robot and the model learning will still be centralized. The use of such a distributed planning approach will decouple the next action selection for each robot, however,

it will also put restrictions on the coordination of robots as the framework can only use one robot in the designated area.

We simulated this distributed planning approach to compare its performance with the suggested centralized framework. We used the same setup as described in section 5.1 for a team of robots. Field 1 and Field 2 were both split into two equal left and right halves for simulating the distributed planning approach. We simulated 10 runs for both the fields and used both the versions of our planning approach. The MAEs for these simulations are shown as boxplots in **Figure 9**. It can be clearly observed from the boxplots that median errors of centralized planning are about half the median errors of distributed planning. This result shows that our centralized planning performs better when compared to the distributed planning.



**FIGURE 8** | Mean absolute error over 10 runs shown as a boxplot for a team of two robots with greedy IPP and *m-AdaPP*. The error calculated in estimating both the fields, Field 1 and Field 2. The greedy IPP is a multi-robot IPP that aims to reduce the maximum sum of variance in the neighboring cells, and hence simulates greedy planning. The error values show that *m-AdaPP* performs better compared to the greedy IPP.



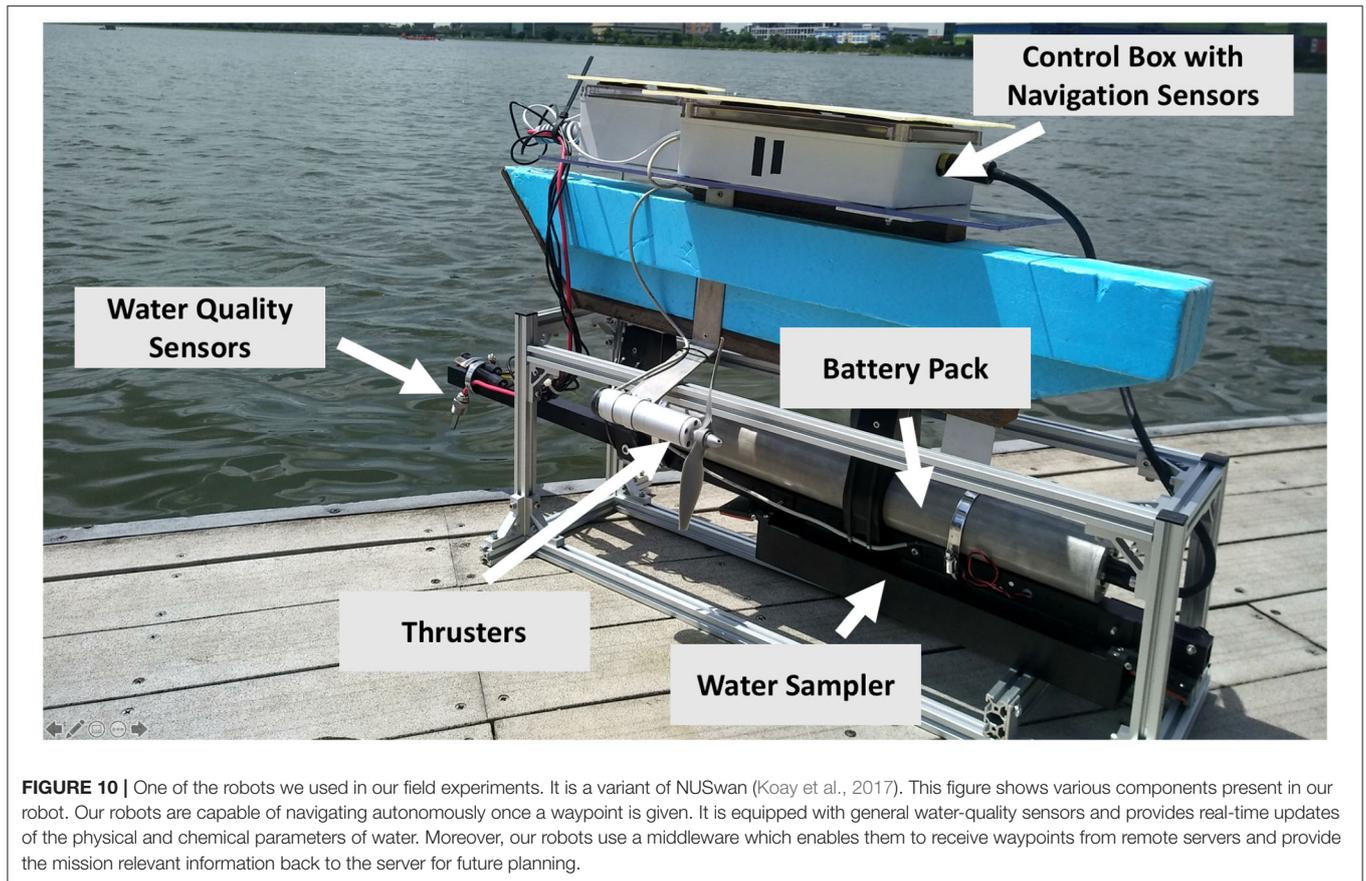
**FIGURE 9** | A boxplot of mean absolute error over 10 runs for a team of two robots with distributed planning and centralized planning of *m-AdaPP*. The result shows the error in estimating both the fields, Field 1 and Field 2. The tiled version of AdaPP for a team of two robots is essentially separating the field into two equal halves and using *m-AdaPP* for planning individually. It is important to note that model learning in distributed planning and centralized planning of *m-AdaPP* is the same and only the approach for planning is different. The error values show that centralized planning performs better when compared to the tiled version.

#### 5.4. Field Experiments for Performance Comparison With Lawn Mower Paths

We tested the performance of our framework against conventional approaches such as estimating fields using lawn mower paths via field experiments. We developed two variants of the NUSwan (Koay et al., 2017) robot as shown in **Figure 10**. These robots were equipped with general water-quality sensors such as DO, conductivity, pH, and oxidation-reduction potential. Moreover, these robots used on-board navigation sensors to guide the robot to the locations given by the framework. Our

framework *m-AdaPP* was hosted on a cloud server, which can be accessed by our robots using a mobile network. This cloud server was a compute instance provided by Amazon Web Service with the capability to run 16 threads in parallel. This capability is crucial for our framework as it significantly reduces the computation time for making planning decisions. We optimized our framework to run smoothly on this compute instance. Both the robots posted the data to this server every 5 s.

For consistency, the mission time for the team of two robots for our field experiments is the same as the mission time we



**FIGURE 10 |** One of the robots we used in our field experiments. It is a variant of NUSwan (Koay et al., 2017). This figure shows various components present in our robot. Our robots are capable of navigating autonomously once a waypoint is given. It is equipped with general water-quality sensors and provides real-time updates of the physical and chemical parameters of water. Moreover, our robots use a middleware which enables them to receive waypoints from remote servers and provide the mission relevant information back to the server for future planning.

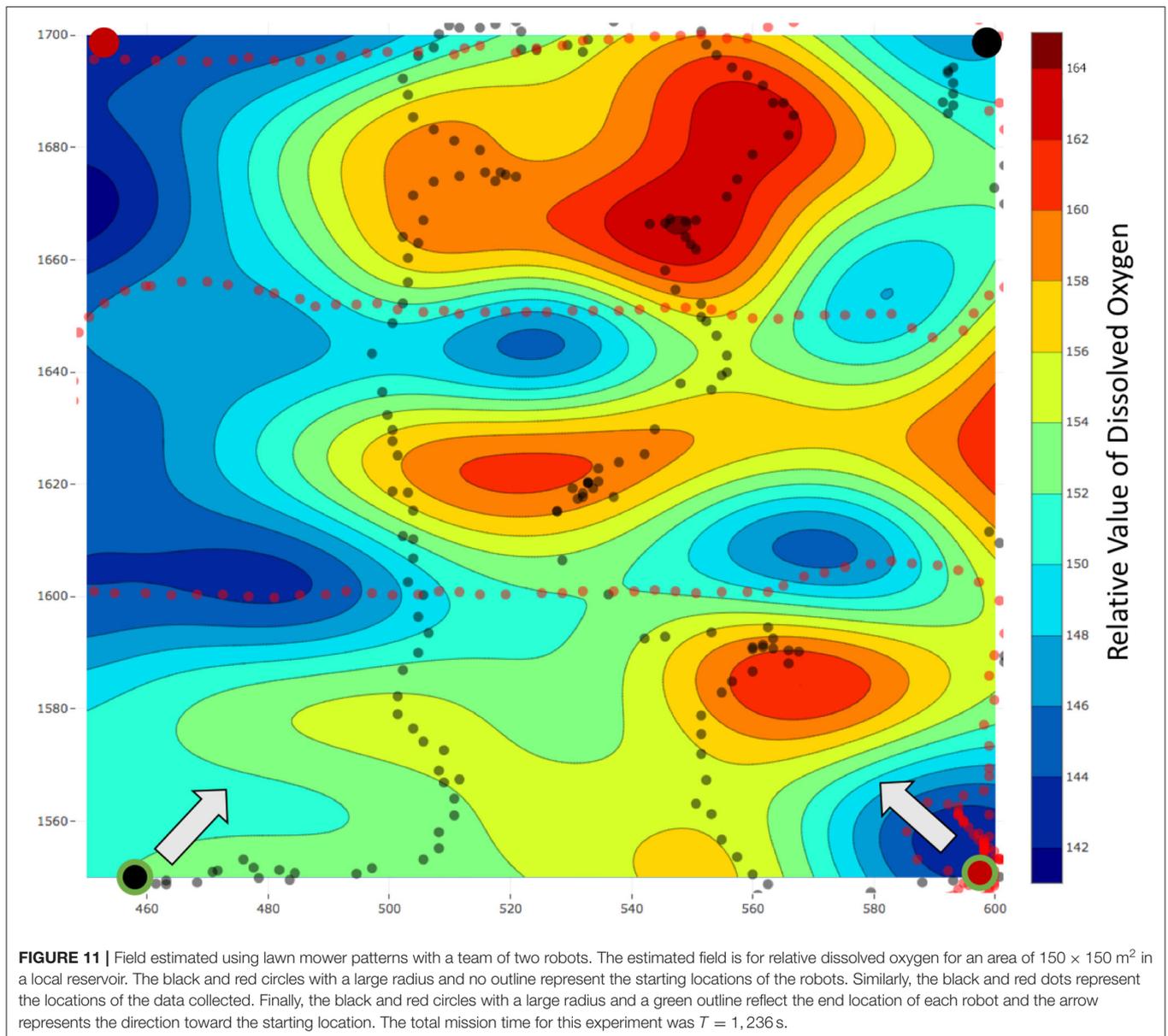
used for two robots in our simulations, which is  $T = 1,200$  s and  $T_s = 120$  s. In general, lawn mower paths are defined by the number of legs, where each leg is a straight path parallel to one of the axes of the survey area. If the speed of the vehicle is constant, lawn mowers can be defined in terms of time but speed of the vehicle in the field can vary due to external disturbances. Therefore, the lawn mowers are defined in terms of lengths rather than time.

Imposing the temporal constraints directly on the lawn mower paths can result in abruptly stopping the lawn mower pattern. Therefore, we assume an average speed of the robots and use this average speed to calculate the total length of the lawn mower for the mission time  $T = 1,200$  s. We set this average speed as 0.5 m/s. Note that this average speed is higher than the average speed mentioned earlier. This difference is to factor in the fact that the vehicle mostly moves in a straight line and thus inertia of the vehicle helps in maintaining a higher speed. Using the average speed of 0.5 m/s and a mission time of  $T = 1,200$  s, we set the length of the lawn mower as 600 m.

We selected a survey field of area  $150 \times 150$  m<sup>2</sup> in a local reservoir and used our robots to estimate the field of DO over this area. The estimated fields using the lawn mower patterns and our frameworks are present in **Figures 11, 12**, respectively. The mission time for the lawn mower paths was 1,236 s and thus our assumption of a higher average speed was correct.

Additionally, the distance traveled by the robots while using our framework is less when compared to lawn mowers, generally within a 5% range. The distance traveled is less mainly due to momentary stops during synchronizations between the team of robots and the server. The black and red circles with a large radius and no outline represent the starting locations of the robots in **Figures 11, 12**, whereas, the circles with a green outline represent the end location of the robots.

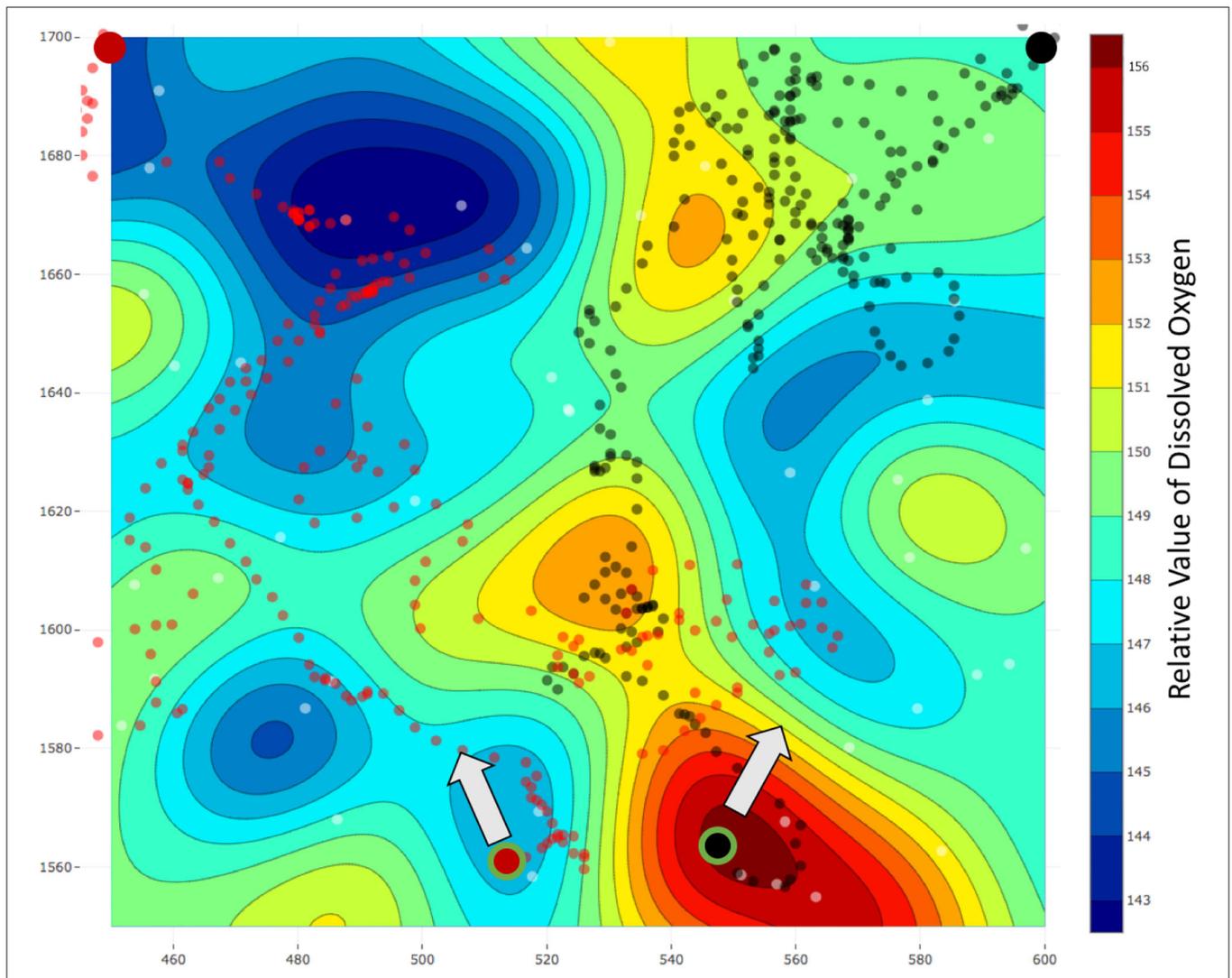
We collected a test dataset to measure the performance of our framework and the lawn mower paths. This test dataset was collected while robots were traveling back to the starting location after finishing the mission. This dataset contained both the locations as well as the ground truth data for the respective locations. We obtained the estimated DO value for these locations using the learned models and calculated the errors using the collected ground truth data. Additionally, we calculated the mean and standard deviation for each of the collected test datasets. These statistical values can be used to approximate the similarity between the two datasets. The results for both are presented in **Table 2**. It can be observed that both the test dataset had similar characteristics and thus the errors of the two methods can be compared. The calculated errors for our framework are significantly lower compared to the errors for the lawn mower paths. These field experiments demonstrate that our framework is able to provide a better estimate of the environmental field.



## 5.5. Using Estimated Fields for Scientific Experiments

The aim of these experiments was to use the estimated fields for selecting locations to collect water samples from different concentrations of a water-quality parameter and use these water samples to understand the micro-level interactions. For the sample collection process, we performed three field estimation tasks using our framework. Two out of these three estimation tasks were on the same day with a temporal difference of 1 h. Each of these estimation tasks were given a mission time of 20 min. The following was the overall schedule of our experiments: 10:30 am on February 28, 2019, 01:05 p.m. on March 4, 2019, and 02:25 p.m. on March 4, 2019. These estimated fields are shown in **Figures 13, 14**, where all the values of dissolved oxygen (DO) are a relative measure of DO instead of the true values.

After each field was estimated, we manually selected the locations to sample and used the robot's automated sampler to collect 1 L of water. These sampling locations are shown as red and blue circles with white outlines in **Figures 13, 14**. In total, we collected three samples from the regions with low DO values (cold regions) and four samples from the regions with high DO values (hot regions). These samples were then sent for lab analysis such as sample filtering, DNA sequencing, and assembling the DNA to identify different microorganisms. We used the PHRED quality score (Ewing and Green, 1998; Ewing et al., 1998) for our samples. This score is a value between 2 and 40 and it is used to check the quality of the samples before performing any further analysis. This value will be low if the amount of information such as the total DNA present in the samples is not enough to construct and identify the microorganisms. Similarly,



**FIGURE 12 |** Field estimated using a team of two robots and our framework *m-AdaPP*. The estimated field is for relative dissolved oxygen for an area of  $150 \times 150$   $m^2$  in a local reservoir. The black and red circles with a large radius and no outline represent the starting locations of the robots. Similarly, the black and red dots represent the locations of the data collected. Finally, the black and red circles with a large radius and a green outline reflect the end location of each robot and the arrow represents the direction toward the starting location. The total mission time for this experiment was  $T = 1,200$  s with  $T_s = 120$  s. It is interesting to observe that data collected using this team of robots were dense in a few regions, whereas, sparse for the remaining regions. However, our framework still performs better as compared to the lawn mower pattern and this is a field-validated result that collecting representative data (adaptive framework) can perform better when compared to collecting data with repetitive information (lawn mowers).

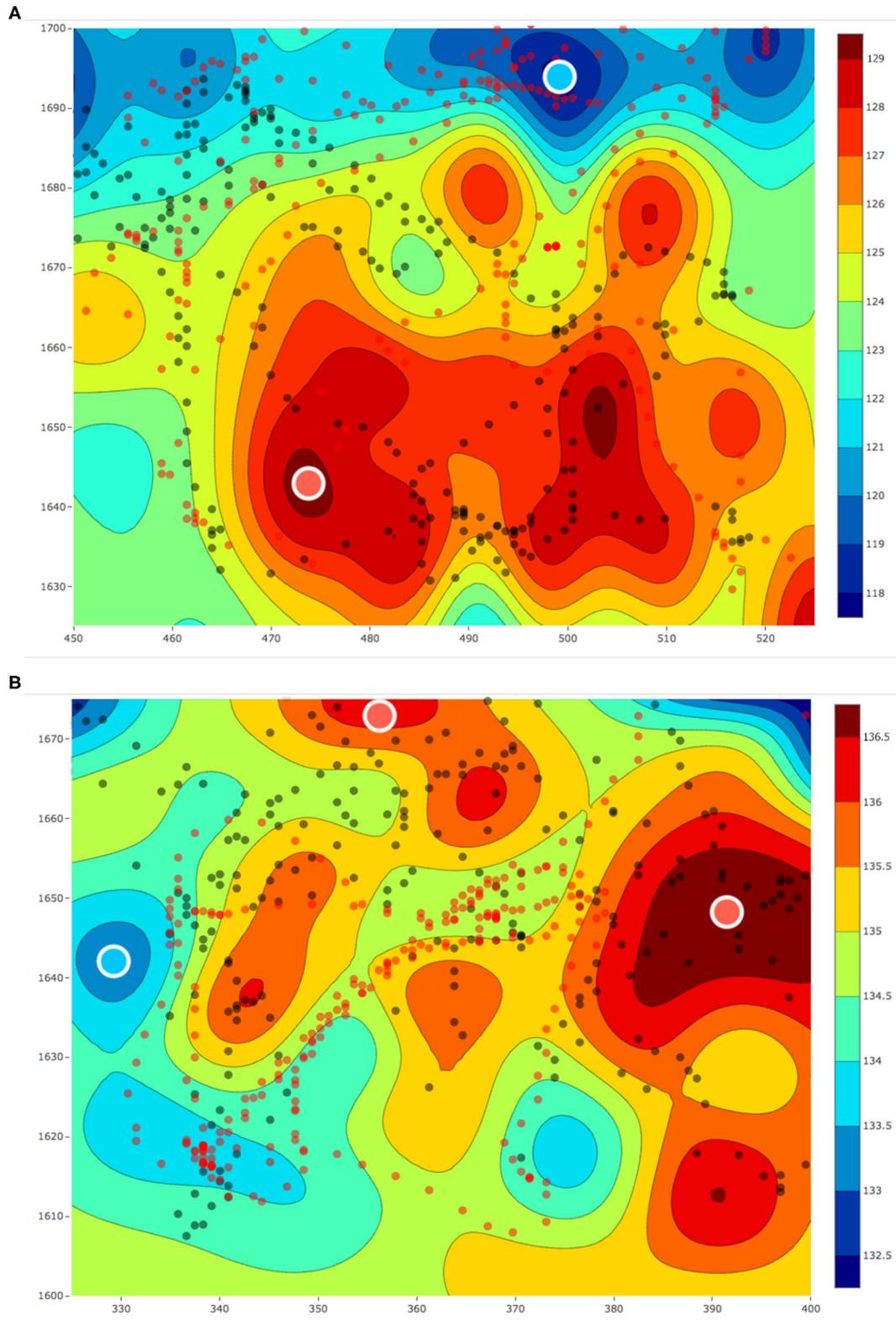
this value will be high if the amount of information present in the collected samples is enough for further analysis such as identifying microorganisms. The PHRED quality score can vary due to many different reasons such as sampling location or the filtering process, and thus having an objective score makes it easier to evaluate the samples collected. The mean scores after denoising was approximately 30.

After our quality analysis, we performed further analysis to find the exact microorganisms present in our samples and examined the differences between hot and cold regions estimated by our framework. **Figure 15** shows the principal

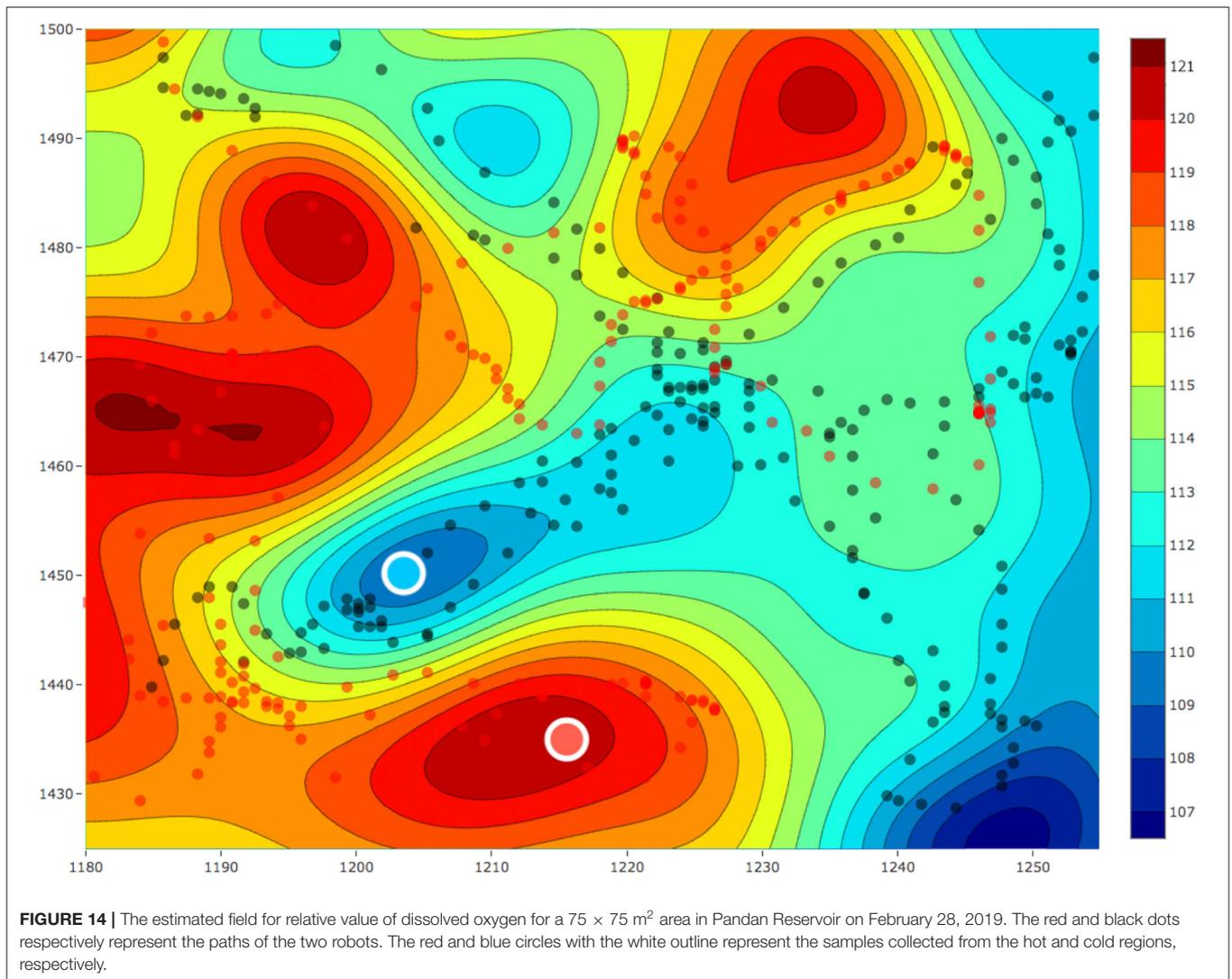
**TABLE 2 |** The root mean square error (RMSE), mean absolute error (MAE), and statistics for the test dataset used for each approach.

| Estimated using     | Estimation error |     | Test data |                |
|---------------------|------------------|-----|-----------|----------------|
|                     | RMSE             | MAE | Mean      | Std. deviation |
| Lawn mower patterns | 6.6              | 4.8 | 148.4     | 5.2            |
| <i>m-AdaPP</i>      | 3.9              | 2.8 | 149.1     | 4.5            |

*These values represent the error in the estimation of the dissolved oxygen field. These errors were calculated using the test data collected by the team of robots while returning to the starting location and the mean and standard deviation for each of the test dataset is available in the last two columns. Our framework gives about a 50% improvement in performance as compared to the fields estimated using lawn mowers.*



**FIGURE 13 | (A)** The estimated fields of relative values of dissolved oxygen for a  $75 \times 75 \text{ m}^2$  area in Pandan Reservoir on March 4, 2019 at **(A)** 01 : 05 p.m. and **(B)** 02 : 25 p.m. The red and black dots, respectively, represent the paths of the two robots. The red and blue circles with the white outline represent the samples collected from the hot and cold regions, respectively.



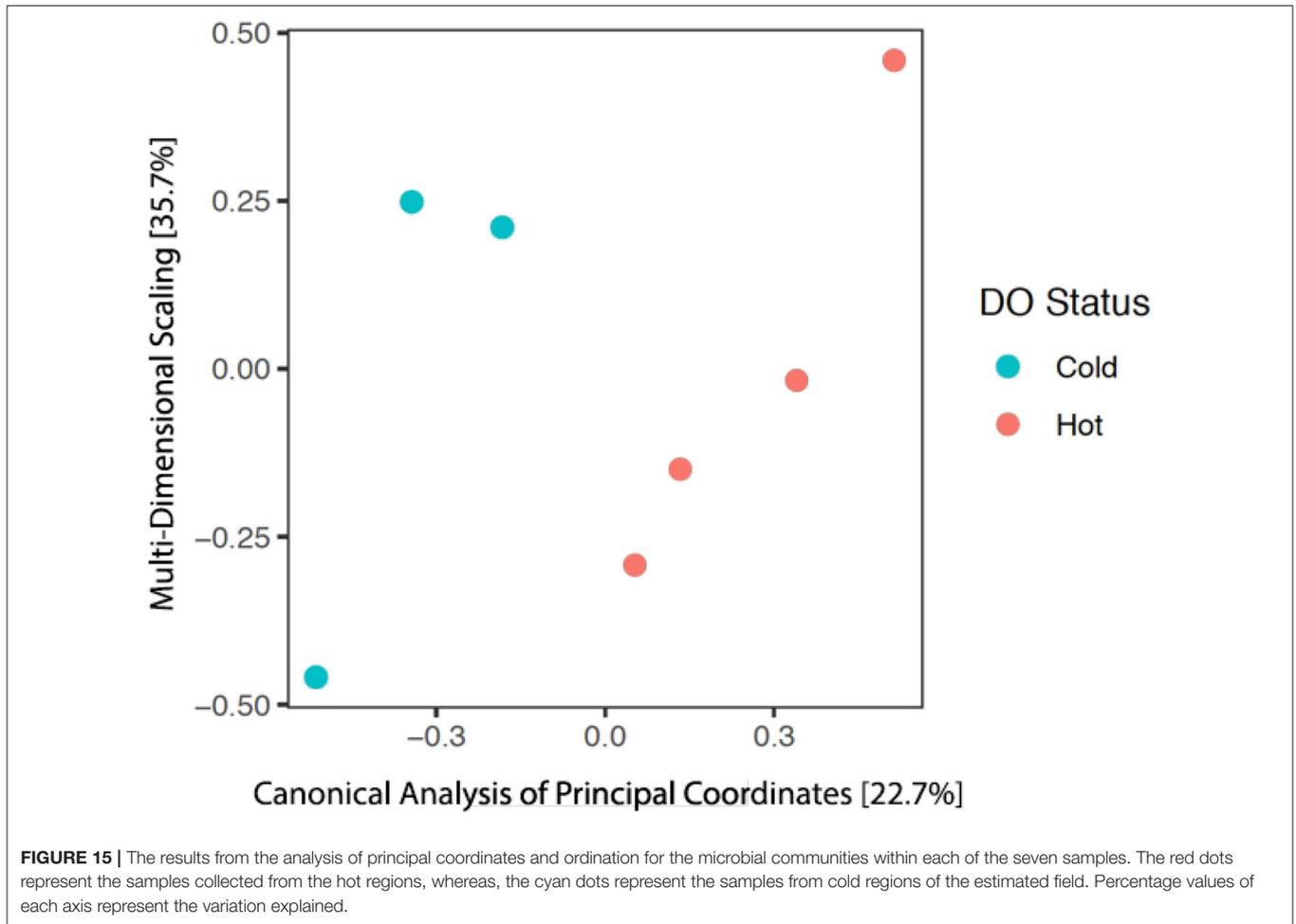
coordinate analysis (PCoA) (Anderson and Willis, 2003), which is a commonly used method to find the dissimilarities between a group of microorganisms in a sample. Although we performed only three experiments, the results shown in **Figure 15** are encouraging. It is clearly evident that the group of microorganisms living in the hot regions are substantially different from the group living in the cold regions of the estimated fields. Therefore, these preliminary results provide a good use case for the adaptive frameworks. Such field estimation experiments can further help in understanding the biological questions such as explaining the difference in the groups of microorganisms between the cold and hot regions.

## 6. CONCLUSION

We outlined a framework for monitoring scalar environmental fields using a team of robots with bounds on overall mission time. We used the kernel information of the sparse GP model

to explore the combinations of actions available to the team of robots and collect informative data. The paths are evaluated to minimize the overall variance and we include the time taken for this evaluation in our overall mission time to provide real-time performance. We simulated the framework using real world data and the results show that our framework is capable of coordinating a team of robots efficiently. We also simulated multiple runs of the framework to test the robustness in our performance and the results show consistent results across multiple simulations.

We designed two robots based on the NUSwan vehicle for monitoring reservoirs in Singapore. Using this team of robots, we validated the performance of our framework in the field against conventional methods such as using lawn mower paths. The estimation error for these field experiments was based on the test data collected after finishing the monitoring task and the results show that our framework outperforms the lawn mower approach. Overall, we explained and validated our contribution for using a team of robots to estimate a scalar environmental field.



We further examined the biological relevance of the fields estimated using our multi-robot framework, *m-AdaPP*. We used the framework to estimate three fields and find the regions of high (hot) and low (cold) concentrations for each survey area. After completing each survey, we collected physical water samples using our robots and used standard scientific protocols to analyze the communities of microorganisms in the samples. These standard lab-based methods were sample filtering, DNA sequencing, and assembling the DNA to identify different microorganisms. The results show the samples collected using our framework are of good quality and can be used for biological studies. Moreover, we analyzed our samples collected from hot and cold regions and found the microorganism communities to be distinct.

## 7. LIMITATIONS AND FUTURE WORK

The suggested *m-AdaPP* framework has two limitations. First, the centralized approach for coordinating the team of robots.

Our framework solves the best actions for the entire team of robots and thus the size of the decision space is directly related to the number of robots. This direct relationship results in high computational cost for a large team of robots. An approach to address this limitation can be a distributed algorithm.

The second limitation comes from the use of SPGP. Although the training time scales with  $NM^2$  instead of  $N^3$  still having a very large number of training points,  $N$  will affect the performance of our framework. A simple solution to this problem will be the use of streaming GPs as the training time as these GP models are completely independent of the training points  $N$ .

The field experiments primarily showed the use for in-water applications. However, the problem formulation of our framework does not put a limitation on the applications and it can be easily extended to estimate any scalar field that can be approximated using GPs. Our framework can be easily used for the estimation of air temperature or estimation of vegetation spread using aerial or land vehicles.

## DATA AVAILABILITY STATEMENT

The dataset used in simulations is available publicly and the citations are available in this paper. The data collected via field experiments is confidential.

## AUTHOR CONTRIBUTIONS

RM contributed in problem formulation, implementation of the framework, simulations, system design, field experiments, and sample collection using the robots. TK contributed in system design and provided supervision for field experiments and sample collection using the robots. MC contributed in problem formulation and provided supervision for implementation of the framework, analysis of simulation results, and system design. SS contributed in managing field experiments, freshwater sample

collection as well as its analysis. All authors contributed to the article and approved the submitted version.

## FUNDING

This work was supported by the Singapore Center for Environmental Life Sciences Engineering (SCELSE) and the National Research Foundation CRP (Water)'s PUB-000-1803-002 grant that is administered by PUB, Singapore's National Water Agency. RM was also thankful to the NUS Graduate School of Integrative Sciences and Engineering for supporting his doctoral study.

## ACKNOWLEDGMENTS

The authors are thankful to Dr. Aditya Bandla from SCELSE for processing and analyzing the samples collected by our robots.

## REFERENCES

- Anderson, M. J., and Willis, T. J. (2003). Canonical analysis of principal coordinates: a useful method of constrained ordination for ecology. *Ecology* 84, 511–525. doi: 10.1890/0012-9658(2003)084[0511:CAOPCA]2.0.CO;2
- Camilli, R., Reddy, C. M., Yoerger, D. R., Van Mooy, B. A., Jakuba, M. V., Kinsey, J. C., et al. (2010). Tracking hydrocarbon plume transport and biodegradation at Deepwater Horizon. *Science* 330, 201–204. doi: 10.1126/science.1195223
- Cao, N., Low, K. H., and Dolan, J. M. (2013). "Multi-robot informative path planning for active sensing of environmental phenomena: a tale of two algorithms," in *AAMAS 2013* (St Paul, MN), 7–14.
- Caron, D. A., Stauffer, B., Moorthi, S., Singh, A., Batalin, M., Graham, E. A., et al. (2008). Macro-to fine-scale spatial and temporal distributions and dynamics of phytoplankton and their environmental driving forces in a small Montane lake in Southern California, USA. *Limnol. Oceanogr.* 53(5 Pt 2), 2333–2349. doi: 10.4319/lo.2008.53.5\_part\_2.2333
- Chen, B., Pandey, P., and Pompili, D. (2012). "An adaptive sampling solution using autonomous underwater vehicles," in *IFAC Proceedings Volume 45*, 352–356. doi: 10.3182/20120919-3-IT-2046.00060
- Darrouzet-Nardi, A., and Bowman, W. D. (2011). Hot spots of inorganic nitrogen availability in an alpine-subalpine ecosystem, Colorado Front Range. *Ecosystems* 14, 848–863. doi: 10.1007/s10021-011-9450-x
- Das, J., Py, F., Harvey, J. B., Ryan, J. P., Gellene, A., Graham, R., Caron, D. A., et al. (2015). Data-driven robotic sampling for marine ecosystem monitoring. *Int. J. Robot. Res.* 34, 1435–1452. doi: 10.1177/0278364915587723
- Das, J., Rajany, K., Frolovy, S., Pyy, F., Ryany, J., Caronz, D. A., et al. (2010). "Towards marine bloom trajectory prediction for AUV mission planning," in *2010 IEEE International Conference on Robotics and Automation (ICRA)* (Anchorage, AK), 4784–4790. doi: 10.1109/ROBOT.2010.5509930
- Deltares (2006). *Delft3d-Flow User Manual*. Available online at: [https://content.oss.deltares.nl/delft3d/manuals/Delft3D-FLOW\\_User\\_Manual.pdf](https://content.oss.deltares.nl/delft3d/manuals/Delft3D-FLOW_User_Manual.pdf)
- Dunabin, M., and Marques, L. (2012). Robots for environmental monitoring: significant advancements and applications. *IEEE Robot. Autom. Mag.* 19, 24–39. doi: 10.1109/MRA.2011.2181683
- Ewing, B., and Green, P. (1998). Base-calling of automated sequencer traces using phred. I. Error probabilities. *Genome Res.* 8, 186–194. doi: 10.1101/gr.8.3.186
- Ewing, B., Hillier, L., Wendl, M. C., and Green, P. (1998). Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. *Genome Res.* 8, 175–185. doi: 10.1101/gr.8.3.175
- Hart, W. E., and Murray, R. (2010). Review of sensor placement strategies for contamination warning systems in drinking water distribution systems. *J. Water Resour. Plann. Manage.* 136, 611–619. doi: 10.1061/(ASCE)WR.1943-5452.0000081
- Hengl, T., Heuvelink, G. B., and Stein, A. (2004). A generic framework for spatial prediction of soil variables based on regression-kriging. *Geoderma* 120, 75–93. doi: 10.1016/j.geoderma.2003.08.018
- Hitz, G., Galceran, E., Garneau, M.-É., Pomerleau, F., and Siegwart, R. (2017). Adaptive continuous-space informative path planning for online environmental monitoring. *J. Field Robot.* 34, 1427–1449. doi: 10.1002/rob.21722
- Hitz, G., Gotovos, A., Garneau, M.-É., Pradalier, C., Krause, A., Siegwart, R. Y., et al. (2014). "Fully autonomous focused exploration for robotic environmental monitoring," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 2658–2664. doi: 10.1109/ICRA.2014.6907240
- Hollinger, G., and Singh, S. (2008). Proofs and experiments in scalable, near-optimal search by multiple robots. *Proc. Robot.* 1, 206–213. doi: 10.15607/RSS.2008.IV.027
- Hosoda, S., Suga, T., Shikama, N., and Mizuno, K. (2009). Global surface layer salinity change detected by Argo and its implication for hydrological cycle intensification. *J. Oceanogr.* 65:579. doi: 10.1007/s10872-009-0049-1
- JPL MUR MEaSURES Project (2010). *GHRSS Level 4 MUR Global Foundation Sea Surface Temperature Analysis*. Available online at: <http://dx.doi.org/10.5067/GHGMR-4FJ01> (accessed January 1, 2018).
- Kemna, S., Rogers, J. G., Nieto-Granda, C., Young, S., and Sukhatme, G. S. (2017). "Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore), 2124–2130. doi: 10.1109/ICRA.2017.7989245
- Koay, T. B., Chitre, M., and Ong, C. N. (2015). *Using Swan to Monitor Water Quality in Reservoirs*. Innovation in water - Research highlights by PUB, 7:15.
- Koay, T. B., Raste, A., Tay, Y. H., Wu, Y., Mahadevan, A., Tan, S. P., et al. (2017). Interactive monitoring in reservoirs using nuswan-preliminary field results. *Water Pract. Technol.* 12, 806–817. doi: 10.2166/wpt.2017.089
- Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* 9, 235–284. Available online at: <http://jmlr.org/papers/v9/krause08a.html>
- Le, N. D., and Zidek, J. V. (2006). *Statistical Analysis of Environmental Space-Time Processes*. Springer Science & Business Media.
- Low, K. H., Chen, J., Dolan, J. M., Chien, S., and Thompson, D. R. (2012). "Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing," in *AAMAS 2012* (Valencia), 105–112. doi: 10.1007/978-1-137-02806-8\_7
- Low, K. H., Dolan, J. M., and Khosla, P. (2008). "Adaptive multi-robot wide-area exploration and mapping," in *AAMAS 2008* (Estoril), 23–30.

- Low, K. H., Dolan, J. M., and Khosla, P. (2011). "Active Markov information-theoretic path planning for robotic environmental sensing," in *AAMAS 2011* (Taipei), 753–760.
- Ma, K.-C., Liu, L., and Sukhatme, G. S. (2017). "Informative planning and online learning with sparse Gaussian processes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore), 4292–4298. doi: 10.1109/ICRA.2017.7989494
- MacKay, D. J. (1998). "Introduction to Gaussian processes," in *NATO ASI Series F Computer and Systems Sciences* (Berlin: Springer), 133–165.
- Meliou, A., Krause, A., Guestrin, C., and Hellerstein, J. M. (2007). "Nonmyopic informative path planning in spatio-temporal models," in *AAAI, Vol. 10* (Vancouver, BC), 16–17.
- Mignot, A., Claustre, H., Uitz, J., Poteau, A., D'Ortenzio, F., and Xing, X. (2014). Understanding the seasonal dynamics of phytoplankton biomass and the deep chlorophyll maximum in oligotrophic environments: a Bio-Argo float investigation. *Glob. Biogeochem. Cycles* 28, 856–876. doi: 10.1002/2013GB004781
- Mishra, R. (2019). *Information based adaptive path planning and sampling for environment monitoring* (Ph.D. thesis). National University of Singapore, Singapore, Singapore.
- Mishra, R., Chitre, M., and Swarup, S. (2018). "Online informative path planning using sparse Gaussian processes," in *OCEANS 2018 MTS/IEEE Conference and Exhibition* (Kobe), 1–5. doi: 10.1109/OCEANSKOBE.2018.8559183
- Moore, A. M., Arango, H. G., Broquet, G., Powell, B. S., Weaver, A. T., and Zavala-Garay, J. (2011). The Regional Ocean Modeling System (ROMS) 4-dimensional variational data assimilation systems: part I-system overview and formulation. *Prog. Oceanogr.* 91, 34–49. doi: 10.1016/j.pocean.2011.05.004
- Newman, K. R., Cormier, M.-H., Weissel, J. K., Driscoll, N. W., Kastner, M., Solomon, E. A., et al. (2008). Active methane venting observed at giant pockmarks along the US mid-Atlantic shelf break. *Earth Planet. Sci. Lett.* 267, 341–352. doi: 10.1016/j.epsl.2007.11.053
- Oh, W., and Lindquist, B. (1999). Image thresholding by indicator kriging. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 590–602. doi: 10.1109/34.777370
- Palta, M. M., Ehrenfeld, J. G., and Groffman, P. M. (2014). "Hotspots" and "hot moments" of denitrification in urban brownfield wetlands. *Ecosystems* 17, 1121–1137. doi: 10.1007/s10021-014-9778-0
- Pascoal, A., Oliveira, P., Silvestre, C., Sebastião, L., Rufino, M., Barroso, V., et al. (2000). "Robotic ocean vehicles for marine science applications: the European Asimov project," in *Oceans 2000 MTS/IEEE Conference and Exhibition, Vol. 1* (Providence, RI), 409–415. doi: 10.1109/OCEANS.2000.881293
- Petillo, S. M. (2015). *Autonomous & Adaptive Oceanographic Feature Tracking on Board Autonomous Underwater Vehicles*. Technical report, Woods Hole Oceanographic Institution MA. doi: 10.1575/1912/7129
- Rasmussen, C. E., and Williams, C. K. (2004). *Gaussian Processes in Machine Learning*. Advanced Lectures on Machine Learning. Springer, 63–71. doi: 10.1007/978-3-540-28650-9\_4
- Roemmich, D., Boebel, O., Desaubies, Y., Freeland, H., King, B., LeTraon, P.-Y., et al. (1999). Argo: the global array of profiling floats. *CLIVAR Exchang.* 13, 4–5.
- Seeger, M., Williams, C., and Lawrence, N. (2003). "Fast forward selection to speed up sparse Gaussian process regression," in *Artificial Intelligence and Statistics, 9* (Key West, FL).
- Singh, A., Krause, A., Guestrin, C., and Kaiser, W. J. (2009). Efficient informative sensing using multiple robots. *J. Artif. Intell. Res.* 34, 707–755. doi: 10.1613/jair.2674
- Siswanto, E., Ishizaka, J., Morimoto, A., Tanaka, K., Okamura, K., Kristijono, A., et al. (2008). Ocean physical and biogeochemical responses to the passage of Typhoon Meari in the East China Sea observed from Argo float and multiplatform satellites. *Geophys. Res. Lett.* 35:L15604. doi: 10.1029/2008GL035040
- Smith, R. N., Pereira, A., Chao, Y., Li, P. P., Caron, D. A., Jones, B. H., et al. (2010). "Autonomous underwater vehicle trajectory design coupled with predictive ocean models: a case study," in *2010 IEEE International Conference on Robotics and Automation (ICRA)* (Anchorage, AK), 4770–4777. doi: 10.1109/ROBOT.2010.5509240
- Smith, R. N., Schwager, M., Smith, S. L., Jones, B. H., Rus, D., and Sukhatme, G. S. (2011). Persistent ocean monitoring with underwater gliders: adapting sampling resolution. *J. Field Robot.* 28, 714–741. doi: 10.1002/rob.20405
- Snelson, E., and Ghahramani, Z. (2006). "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 1257–1264.
- Stanev, E. V., He, Y., Staneva, J., and Yakushev, E. (2014). Mixing in the Black Sea detected from the temporal and spatial variability of oxygen and sulfide-Argo float observations and numerical modelling. *Biogeosciences* 11, 5707–5732. doi: 10.5194/bg-11-5707-2014
- Stein, M. L. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media.
- Sukhatme, G. S., Dhariwal, A., Zhang, B., Oberg, C., Stauffer, B., and Caron, D. A. (2007). Design and development of a wireless robotic networked aquatic microbial observing system. *Environ. Eng. Sci.* 24, 205–215. doi: 10.1089/ees.2006.0046
- Webster, R., and Oliver, M. (2007). *Geostatistics for Environmental Scientists*. John Wiley & Sons Inc. doi: 10.1002/9780470517277
- YSI (2017). *Multi-Parameter Water Quality Meter: YSI 556 MPS*. Available online at: <http://store.eonpro.com/store/p/1814-Multi-Parameter-Water-Quality-Meter-YSI-556-MPS.aspx>
- Yu, J., Schwager, M., and Rus, D. (2016). Correlated orienteering problem and its application to persistent monitoring tasks. *IEEE Trans. Robot.* 32, 1106–1118. doi: 10.1109/TRO.2016.2593450
- Zhang, B., and Sukhatme, G. S. (2007). "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *2007 IEEE International Conference on Robotics and Automation* (Roma), 3673–3680. doi: 10.1109/ROBOT.2007.364041
- Zhang, Y., Ryan, J. P., Bellingham, J. G., Harvey, J. B., and McEwen, R. S. (2012). Autonomous detection and sampling of water types and fronts in a coastal upwelling system by an autonomous underwater vehicle. *Limnol. Oceanogr.* 10, 934–951. doi: 10.4319/lom.2012.10.934
- Zhu, G., Wang, S., Wang, W., Wang, Y., Zhou, L., Jiang, B., et al. (2013). Hotspots of anaerobic ammonium oxidation at land-freshwater interfaces. *Nat. Geosci.* 6:103. doi: 10.1038/ngeo1683
- Zhu, G., Wang, S., Zhou, L., Wang, Y., Zhao, S., Xia, C., et al. (2015). Ubiquitous anaerobic ammonium oxidation in inland waters of China: an overlooked nitrous oxide mitigation process. *Sci. Rep.* 5:17306. doi: 10.1038/srep17306

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Mishra, Koay, Chitre and Swarup. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.