Online Informative Path Planning using Sparse Gaussian Processes

Rajat Mishra*, Mandar Chitre*† and Sanjay Swarup‡
*NUS Graduate School of Integrative Sciences and Engineering, National University of Singapore, Singapore
†Faculty of Engineering, National University of Singapore, Singapore
‡Department of Biological Sciences, National University of Singapore, Singapore
e-mail: rajat.mishra@u.nus.edu, {mandar, sanjay}@nus.edu.sg

***Abstract*—Estimating the environmental fields for large survey areas is a difficult task, primarily because of the field's spatio-temporal nature. A good approach in performing this task is to do adaptive sampling using robots. In such a scenario, robots have limited time to collect data before the field varies significantly. In this paper, we suggest an algorithm, AdaPP, to perform this task of data collection within a constraint on sampling time and provide an approximation of the environmental field. We test our performance against conventional sampling paths and show that we are able to obtain a good approximation of the field within the stipulated time.**

I. INTRODUCTION

The environmental processes taking place in oceans, lakes, rivers, and other water resources are of interest to a broad range of marine and environmental scientists. These processes typically span across large areas and exhibit spatio-temporal variations, which makes the task of environmental monitoring challenging. Increasingly, this task is being automated using Autonomous Underwater Vehicles (AUVs) [1], which is an advancement over the use of static sensor buoys for monitoring. In a simple monitoring task, the AUV traverses a predefined path, generally a lawn mower pattern, for data collection and uses this collected data to provide an estimate of the field. However, such an approach is sub-optimal as the AUV follows a predefined path without benefiting from the incoming information, instead of adjusting its path to get a better approximation of the field.

We are interested in obtaining a good estimate of scalar environmental fields, such as chlorophyll concentration, conductivity or temperature. This can be achieved by collecting representative data of the spatio-temporal field using robots. In general, the temporal changes in environment occur over a period of days or months [2, 3, 4]. Therefore, it is safe to assume that the environmental processes are temporally static for a period of a few minutes. Such an assumption can be useful for reducing the modeling complexity of the environmental field. However, this assumption will also impose a time constraint on the data collection process. In the literature, the path planning algorithms that solve the problem of data collection are commonly termed as Informative Path Planning (IPP) algorithms [5, 6, 7].

The IPP algorithms can be broadly classified as adaptive IPP (online) and non-adaptive IPP (offline) based on the criteria for generating paths. In the latter, a path is generated at the start of the mission using the prior information about the field. Subsequently, the robot traverses this pre-planned path and provides an estimate of the field at the end of the mission. In contrast, the adaptive IPP algorithms update the pre-planned path on-the-fly, by using the information collected while traversing. This is generally executed in 3 stages, which involve approximating the field using the collected data, generating the next waypoints based on the current approximation and collecting the data while traversing. Since waypoint generation happens on-the-fly after incorporating the recent information, adaptive IPP algorithms are more suited for estimating environmental fields compared to non-adaptive algorithms. We briefly discuss some of the adaptive IPP algorithms here.

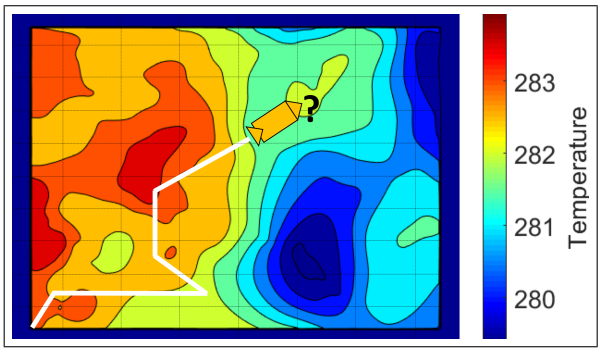

Fig. 1. A scenario similar to the transect sampling task presented in [8]. The environmental field here is the sea surface temperature of an area in Sea of Japan on January 21, 2018. Taken from MUR SST dataset [9].

An entropy-based method for environmental monitoring using multiple robots was proposed [8], where the set of waypoints were obtained by selecting locations of maximum entropy. However, this framework was designed for transect sampling tasks where robots can only move in forward direction. Recently, another adaptive IPP algorithm was suggested to solve the problem of long term monitoring of scalar environmental fields [6]. In this algorithm, the temporal changes were incorporated by re-estimating the field's parameters using heuristic methods. However, in both of these methods, a performance comparison with the conventional lawn mower paths is missing.

One of the critical decisions in the IPP frameworks is to select a good function approximation method to estimate the environmental field. A commonly used approximation method

is Gaussian Processes (GPs) [10, 11]. One of the drawbacks of this method is the exponential scaling of the computation cost with the increasing training dataset. However, sparse GP frameworks overcome this problem by using only a subset of the data to provide accurate estimates. A state-of-the-art sparse GP variant is the Sparse Pseduo-inputs Gaussian Process (SPGP) [12]. Instead of selecting a subset based on some information criteria [13, 14], the SPGP framework learns a pseudo subset that best summarizes the training data.

We present an online IPP framework *AdaPP* for predicting a scalar environmental field. This framework uses a sparse GP method, SPGP, to get an estimate of the field and the corresponding variance in this estimate. The latter statistics, along with the remaining mission time, is used for generating next waypoints. We test our performance on a sea surface temperature dataset and benchmark our results with those from lawn mower paths. The results show that the predicted field using *AdaPP* is a good approximation of the ground truth.

II. BACKGROUND

In this section, we briefly discuss the GP models and the SPGP framework.

A. Sparse Pseudo-inputs Gaussian Processes

GP models are generally used for non-parametric regression problems [15], such as image thresholding [16] and spatial data modeling [17]. In a standard GP model with zero mean prior [15], the kernel function solely models the mapping between input vectors, $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, and target values, $\mathbf{y} = \{y_n\}_{n=1}^N$. For a kernel function $K(\mathbf{x}_n, \mathbf{x}_{n'})$ parameterized by θ, the distribution for a new input $\mathbf{x}$ can be given by:

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

where $[\mathbf{k}_\mathbf{x}]_n = K(\mathbf{x}_n, \mathbf{x})$, $[\mathbf{K}_N]_{n,n'} = K(\mathbf{x}_n, \mathbf{x}_{n'})$, $K_{\mathbf{x},\mathbf{x}} = K(\mathbf{x}, \mathbf{x})$, σ^2 is the noise variance and $\mathcal{D}$ is the training dataset. Therefore, assuming the hyperparameters of the kernel function are known, the predictive function is effectively parameterized by $\mathcal{D}$. As SPGP uses a subset of the training data, this parameterization is done using the pseudo dataset $\overline{\mathcal{D}}$ of size $M \ll N$, which has pseudo-inputs $\overline{\mathbf{X}} = \{\overline{\mathbf{x}}_m\}_{m=1}^M$ and corresponding pseudo targets $\overline{\mathbf{f}} = \{\overline{f}_m\}_{m=1}^M$. The pseudo targets are denoted as $\overline{\mathbf{f}}$ instead of $\overline{\mathbf{y}}$ because these targets are not measured values and therefore, adding the noise variance σ^2 is not required. However, the actual prediction distribution has the noise variance and is given as

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

where $[\mathbf{K}_M]_{mm'} = \mathbf{K}(\overline{\mathbf{x}}_m, \overline{\mathbf{x}}_{m'})$ and $[\mathbf{k}_x]_m = K(\overline{\mathbf{x}}_m, \mathbf{x})$, for m = $1, 2..., M$. The reduction in computation burden for matrix inversion can be clearly observed by comparing 1 and 2, 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 [12], the predictive distribution given a new input $\mathbf{x}_*$ is:

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

where

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

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

The derivation of $\mathbf{Q}_M$ is omitted here for brevity but it is provided in detail in [12]. With spatial data as input, μ_* will represent the mean predicted field and the variance σ_*^2 will contribute to the variance in this prediction. Moreover, the scalar environmental fields can be non stationary [8]. Interestingly, upto a certain extent, SPGP framework is capable of modeling non-stationary GPs by learning the pseudo-inputs from the training data, giving it an edge over other sparse GP methods.

III. TECHNICAL APPROACH

Our aim is to adaptively sample a survey area within a fixed amount of time T, using only a single robot. This problem is **NP**-hard. Therefore, we developed an adaptive sampling algorithm, named as *AdaPP*, to efficiently sample the search space and get a good approximation of the scalar environmental field within the limited mission time.

A. Decomposition of Survey Area

The survey area is generally decomposed into smaller sub-regions to improve the run time of a path planning algorithm. This decomposition is commonly defined using some criteria to get a good representation of the overall field. One such approach, *GreedySubset*, is discussed in [18], where each sub-region is formed using some prior information. However, in most of the environmental monitoring tasks, the prior information is not available and therefore this method is not applicable.

Our approach for spatial decomposition is both simple and computationally fast. We use the distribution of variance in prediction for decomposing the survey area into smaller regions. Our survey area is discretized into C cells, given by $\mathcal{C} = \{c_1, c_2...c_C\}$. The representative point x_{c_b} and variance σ_{*,c_b} of a cell c_b containing Z spatial points are given as:

$$\sigma_{*,c_b}^2 = \frac{\sum_{z=1}^{Z}\sigma_{*,z}^2}{Z}, \quad (6)$$

$$\mathbf{x}_{c_b} = \frac{\sum_{z=1}^{Z}\sigma_{*,z}^2\mathbf{x}_z}{Z}, \quad (7)$$

where $\sigma_{*,z}^2$ and $\mathbf{x}_z$ are the variance and location of a spatial point z. These quantities represent the mean uncertainty and uncertainty weighted average for all locations in a cell c_b. As 6 and 7 are are easy to compute, the survey area can be decomposed into representative sub-regions at a faster rate.

B. Adaptive Path Planning

The main constraint on our planning is the limited mission time. Moreover, as we do not assume any prior information, we use the variance in our prediction at each planning iteration to select the next waypoint. This selection of next waypoint at each planning iteration represents the adaptive nature in our path planning algorithm. Additionally, this selection should be coordinated in such a way that the data collected during the limited mission time should best represent the scalar environmental field. Therefore, this step of obtaining the next waypoints is critical for efficient adaptive sampling and our algorithm *AdaPP* provides an efficient framework for this selection, governed by the variance in current prediction and the remaining mission time. In order to explain our overall framework, we first discuss a simple single robot algorithm with no constraints on planning. Later on, we explain the planning framework of our *AdaPP* algorithm in detail.

1) Single-robot algorithm with no constraints: In a scenario, where the environmental field does not vary temporally, the task of selecting next waypoints is simple. Given the robot's current location $\mathbf{x}_c^1$ and spatially decomposed survey area, the next waypoint can be obtained using policy iteration:

$$V(\mathbf{x}_c) := \max_{a \in \mathbf{A}_{\mathbf{x}_c}} [R(\mathbf{x}_c, a) + \gamma V(\mathbf{x}_{c'})], \tag{8}$$

$$\pi(\mathbf{x}_c) := \arg\max_{a \in \mathbf{A}_{\mathbf{x}_c}} V(\mathbf{x}_c), \tag{9}$$

where

$$R(\mathbf{x}_c, a) = \frac{\sigma^2_{*,c'}}{||\mathbf{x}_c - \mathbf{x}_{c'}||},$$

$\mathbf{x}_{c'}$ is the next cell on taking the action a in a cell $\mathbf{x}_c$, $\mathbf{A}_{\mathbf{x}_c}$ is a set of all possible actions in cell $\mathbf{x}_c$ and γ is the discounting factor. Dynamic Programming (DP) can be used to obtain the optimal policy $\pi^*(\cdot)$ and next the waypoint can be obtained using $\pi^*(\mathbf{x}_c^1)$. In this approach, the diagonal movements are penalized in the reward function $R(\mathbf{x}_c, a)$ to optimize the usage of time and energy, and no constraints are imposed on the single robot planning. We name this algorithm as single-robot DP.

2) Time constraints on planning: As mentioned before, the single-robot DP algorithm works for temporally static environmental fields. In practice, such assumptions are valid only for short intervals, therefore ignoring the temporal constraint completely will not produce representative environmental fields. Hence it is necessary to include temporal constraint on environmental monitoring tasks.

Introducing time constraint to the single-robot DP algorithm is not straightforward. The selection of next waypoint in temporally constrained planning frameworks has to be based on the current variance in prediction as well as the remaining time. Therefore, the optimal action a_*^t at time t can be given as

$$a_*^t = \arg\max_{a^{t^+} \in \mathbf{A}_{\mathbf{x}_c^t}} [U(a^{t^+}) + \eta \vartheta_{T-t}(a^{t^+})], \tag{10}$$

Algorithm 1: *AdaPP* - single-robot Adaptive Path Planning

Data: Starting point ($\mathbf{x_c^t}$), Total mission time (T), SPGP Parameters (M)
Result: Predicted Field $\mathbf{m}_*$
/* Initialization */
Run sampling decomposition using resolution r and $\sigma_*^2 = \kappa$;
/* Algorithm Loop */
while $t < T$ **do**
  Construct the set $\mathbf{A}_{\mathbf{x}_c^t}$;
  for $a^{t^+} \in \mathbf{A}_{\mathbf{x}_c^t}$ **do**
    Estimate $\vartheta_{T-t}(a^{t^+})$ by simulating planning using *single-robot DP*;
    Calculate and store $U(\mathbf{s}'_t) + \eta\vartheta_{T-t}(a^{t^+})$;
  Use 10 to get a_*^t;
  Take the action a_*^t and collect training data $\mathcal{D}$;
  Update the time t;
  θ = Full-GP using $[\mathbf{y}(1:M), \mathcal{X}(1:M)]$;
  $\overline{\mathcal{X}} = \mathcal{X}(rand(M))$;
  $[m_*, \sigma_*^2]$:= Run SPGP($\mathbf{y}$, $\mathcal{X}$, θ, $\overline{\mathcal{X}}$);
  Run sampling decomposition using resolution r and σ_*^2;

where $U(a_{t^+})$ is a function that gives the variance of the cell that will be visited on taking action a^{t^+}, η is a discounting factor, $\mathbf{A}_{\mathbf{x}_c^t}$ is a set of all the possible actions for a robot in cell c and $\vartheta_{T-t}(a^{t^+})$ represents the potential of reducing uncertainty with the remaining time $T-t$, if the action a^{t^+} is taken. In an ideal scenario, we should plan till the end of horizon and select the next waypoint at time t from the path that reduces the maximum uncertainty. This would result in a large tree search problem with each node represented by a tuple of the robot's future location and the remaining time $T-t$, where the branching factor for each node is defined by $\mathbf{A}_{\mathbf{x}_c^t}$. One approach to solve this search problem is to use value function approximation and learn a model for $\vartheta_{T-t}(\cdot)$, however, doing this in an online manner with fewer data points may not deliver good results. Moreover, we are only interested in an estimate of $\vartheta_{T-t}(\cdot)$ to obtain the next waypoint and not in the true value.

In order to obtain an estimate of $\vartheta_{T-t}(\cdot)$, we simulate the future planning iterations using the current distribution of variance and the remaining time. We first start with a set of actions at the current time t given by $\mathbf{A}_{\mathbf{x}_c^t}$. Assuming the time taken to reach the next waypoint is t^1, the next set of actions can be given by $\mathbf{A}_{\mathbf{x}_c^{t^1}}$. Following this approach, each subsequent planning iteration will have $\mathbf{A}_{\mathbf{x}_c^{t^2}}$, $\mathbf{A}_{\mathbf{x}_c^{t^3}}$ and so on till the end of horizon. This will dramatically increase the search space and will not be solvable within a stipulated time. In order to overcome this problem, we make use of the single-robot DP algorithm to selectively traverse our search space and compute an estimate of $\vartheta_{T-t}(\cdot)$.

Moreover, we make an assumption that when a robot visits

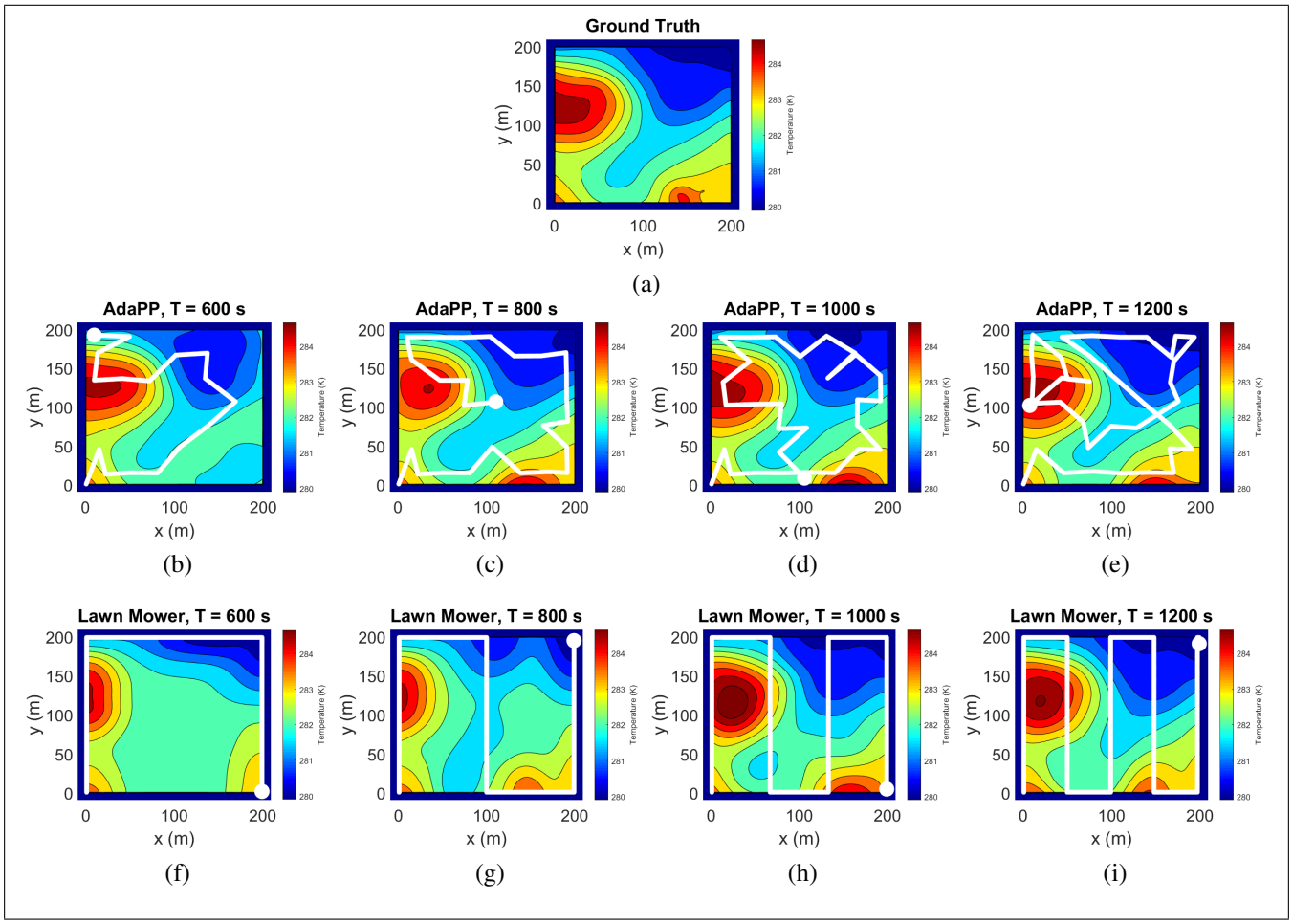

Fig. 2. Simulation results for a 200 m $\times$ 200 m sea surface temperature field. (a) is the ground truth obtained using the MUR dataset [9]. (b)-(e) represents the field predicted using *AdaPP* for different $T = \{600\text{ s}, 800\text{ s}, 1000\text{ s}, 1200\text{ s}\}$ respectively. Similarly, (f)-(i) is the field predicted using lawn mower paths.

a cell during the future planning iterations, the uncertainty in that cell is reduced to the noise variance σ^2. This assumption is only valid if the cell size is substantially smaller than the survey area, which is a common practice in path planning algorithms. Using this assumption, we select one action from the set $\mathbf{A}_{\mathbf{x}_c^t}$ and update the variance of the resulting cell to σ^2. We also update the remaining time by reducing the time taken for completing the selected action. After this, we use single-robot DP algorithm to move to the next best waypoint and update the variance and remaining time again. We repeat this till the end of horizon for all the actions in $\mathbf{A}_{\mathbf{x}_c^t}$ and get an estimate of $\vartheta_{T-t}(\cdot)$ at the current time t. Once the estimates of all the actions at current time are available, the robot can move to the next waypoint obtained using 10.

C. SPGP for predicting the field

A commonly used kernel function in geostatistics is the *automatic relevance determination*, which is defined by $K(\cdot,\cdot)$:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = \alpha \exp\left[\frac{1}{2}\sum_{l=1}^{2} b_l (x_{n,l} - x_{n',l})^2\right] \tag{11}$$

where α, 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 l dimension of $\mathbf{x}_n$. After including the noise term σ^2, the hyperparameters of the sparse GP are given by $\theta = \{\alpha, b_1, b_2, \sigma^2\}$ and pseudo-inputs $\overline{\mathbf{X}}$. These hyperparameters can be learned by maximizing the marginal likelihood as mentioned in [12].

IV. SIMULATION RESULTS

For our simulations, we used the real field Sea Surface Temperature (SST) data of the Sea of Japan provided by the Jet Propulsion Laboratory [9]. We extracted the temperature data for two areas of 200 km $\times$ 200 km each and mapped it to two areas of 200 m $\times$ 200 m each. This mapping was done to conserve the features of an environmental field and have an area that can be explored within a practical value of T.

We simulated the fields for four different static periods, $T = \{600s, 800s, 1000s, 1200s\}$, which are also the constraints on the mission time. The vehicle is assumed to be traveling at constant speed of 1 m/s. For a fair comparison, the four static periods were also used to generate the four different the lawn mower paths, assuming constant vehicle speed. The

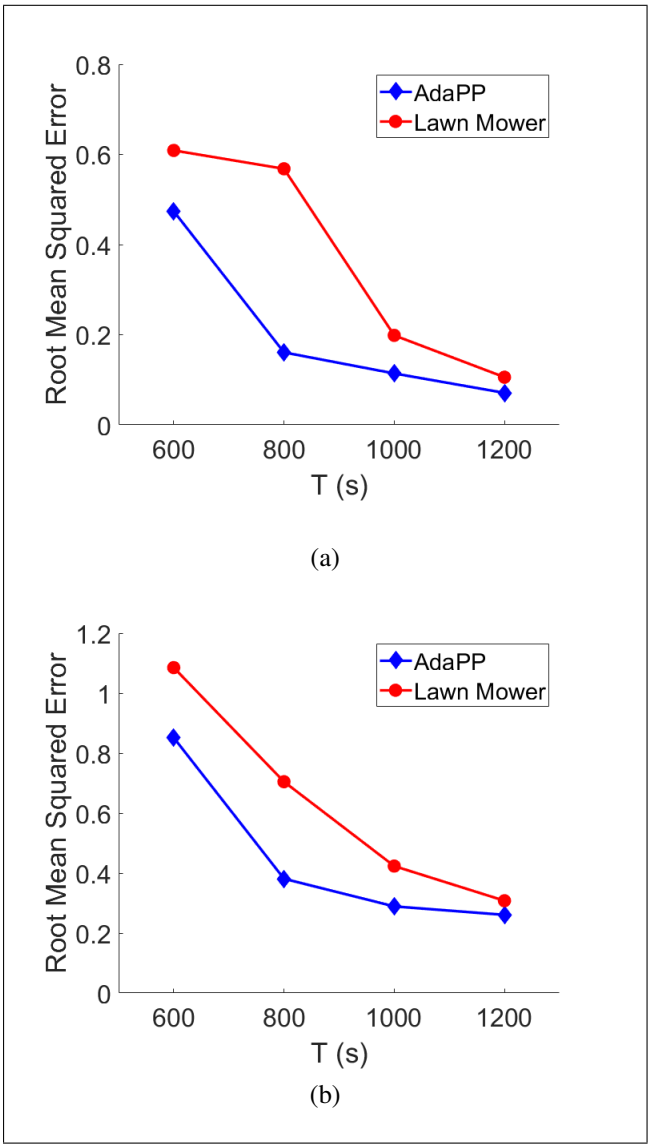

Fig. 3. Root Mean Squared Error in predicting the SST field for two different areas with respect to increasing static period $T(s)$.

total number of pseudo-input points were kept at $M = 50$ and the initial guess for SPGP kernel parameters were obtained by running a full GP regression on a subset of M points.

We implemented our algorithm in MATLAB. For implementation of SPGP, we used the code provided by the authors [12] and modified it for our spatial regression application. The simulations were done on a hexa-core Intel i7 processor with 32 GB of RAM. The results of our simulations for one of the fields are shown in Fig. 2. It is clear from the results that the field predicted using *AdaPP* algorithm presents a better match of the ground truth compared to the field predicted using lawn mower paths. Moreover, the approximation improves with increasing static period T, which is an expected result as the *AdaPP* algorithm has more time to collect data. The trend of root mean square error (RMSE) for the two fields generated using both lawn mower paths and our algorithm are shown in Fig. 3. In all the scenarios, our algorithm has a low RMSE value compared to the lawn mower paths.

V. CONCLUSIONS

In this work, we presented a framework for predicting the scalar environmental field using a single robot with the constraints on mission time. The environmental field was modeled with a sparse GP framework, SPGP, using only a subset of the total training data. The simulations were conducted using real world temperature data and the results show that our algorithm provides a better approximation of the environmental field compared to traditional lawn mower paths.

ACKNOWLEDGMENT

The authors would like to thank Ahmed Mahmood and Hari Vishnu for the productive discussions on GP models. The research work of Rajat Mishra was supported by NUS Graduate School and Singapore Center for Environmental Life Sciences Engineering (SCELSE).

REFERENCES

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.
[2] S. C. Chapra, *Surface water-quality modeling*. Waveland, 2008.
[3] T. G. Sanders and D. D. Adrian, "Sampling frequency for river quality monitoring," *Water Resources Research*, vol. 14, no. 4, pp. 569–576, 1978.
[4] J. C. Loftis and R. C. Ward, "Water quality monitoring some practical sampling frequency considerations," *Environmental Management*, vol. 4, no. 6, pp. 521–526, 1980.
[5] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Robotics and automation (icra), 2010 IEEE international conference on*, IEEE, 2010, pp. 4791–4796.
[6] L. L. Kai-Chieh Ma and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *ICRA*, IEEE, 2017.
[7] G. S. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B. Stauffer, and D. A. Caron, "Design and development of a wireless robotic networked aquatic microbial observing system," *Environmental Engineering Science*, vol. 24, no. 2, pp. 205–215, 2007.
[8] N. Cao, K. H. Low, and J. M. Dolan, "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms," in *AAMAS 2013*, 2013, pp. 7–14.
[9] JPL MUR MEaSUREs Project, *Ghrsst level 4 mur global foundation sea surface temperature analysis*, Dataset accessed [2018-01-30] at http://dx.doi.org/10.5067/GHGMR-4FJ01, 2010.
[10] N. D. Le and J. V. Zidek, *Statistical analysis of environmental space-time processes*. Springer Science & Business Media, 2006.
[11] R. Webster and M. Oliver, *Geostatistics for Environmental Scientists*. John Wiley & Sons Inc, 2007.
[12] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.
[13] L. Csató and M. Opper, "Sparse on-line gaussian processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.
[14] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse gaussian process regression," in *Artificial Intelligence and Statistics 9*, 2003.
[15] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*, Springer, 2004, pp. 63–71.
[16] W. Oh and B. Lindquist, "Image thresholding by indicator kriging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 590–602, 1999.
[17] M. L. Stein, *Interpolation of spatial data: Some theory for kriging*. Springer Science & Business Media, 2012.
[18] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.