

# Spatiotemporal Aquatic Field Reconstruction Using Robotic Sensor Swarm

Yu Wang\*, Rui Tan\*, Guoliang Xing\*, Xiaobo Tan<sup>†</sup>, Jianxun Wang<sup>†</sup>, Ruogu Zhou\*

\*Department of Computer Science and Engineering, Michigan State University, East Lansing, USA

<sup>†</sup>Department of Electrical and Computer Engineering, Michigan State University, East Lansing, USA

Email: {wangyu3, tanrui, glxing, xbtan, wangji19, zhouruog}@msu.edu

**Abstract**—Monitoring important aquatic processes like harmful algal blooms is of increasing interest to public health, ecosystem sustainability, marine biology, and aquaculture industry. This paper presents a novel approach to spatiotemporal aquatic field reconstruction using inexpensive, low-power, mobile sensing platforms called *robotic fish*. Robotic fish networks are a typical example of Cyber-Physical Systems where the design of cyber components (sensing, communication, and information processing) must account for inherent physical dynamics of the robots and the aquatic environment. Our approach features a *rendezvous-based* mobility control scheme where robotic fish collaborate in the form of a *swarm* to sense the aquatic environment in a series of carefully chosen rendezvous regions. We design a novel feedback control algorithm that maintains the desirable level of wireless connectivity for a sensor swarm in the presence of significant environment and system dynamics. Information-theoretic analysis is used to guide the selection of rendezvous regions so that the spatiotemporal field reconstruction accuracy is maximized subject to the limited sensor mobility. The effectiveness of our approach is validated via implementation on sensor hardware and extensive simulations based on real data traces of water surface temperature field and on-water ZigBee wireless communication.

**Keywords**—Robotic sensor swarm, field reconstruction, connectivity control, movement scheduling

## I. INTRODUCTION

Monitoring aquatic environment is of great interest to public health, ecosystem sustainability, marine biology, and aquaculture industry. In this work, we explore an important problem in aquatic monitoring – reconstruction of spatiotemporal aquatic process. Many physical and biological phenomena in aquatic environment, including harmful algal blooms (HABs) [1], lake surface temperature [2], and plume concentration of chemical substance [3], can be modeled as spatiotemporal aquatic fields that usually follow certain distribution such as the spatiotemporal Gaussian process. For instance, Fig. 1(a) shows the HABs in two inland lakes in Wisconsin, 1999. The reconstructed aquatic field allows one to study fine-grained spatial distribution and temporal evolution of physical and biological phenomena of interest. For instance, the reconstructed HAB field is helpful for understanding the development of emerging HABs and guiding authorities to take future preventive actions.

Manual sampling, via boat/ship or with handheld devices, is still a common practice in monitoring aquatic environment. This approach is labor-intensive and has difficulty

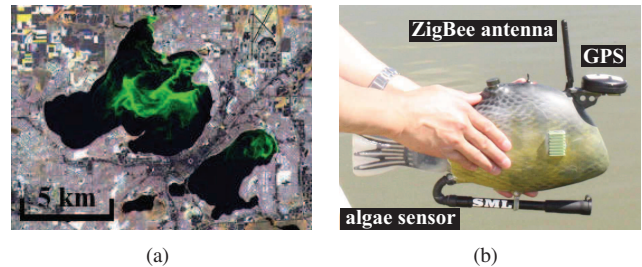


Figure 1. (a) HABs in Lake Mendota (top left) and Lake Monona (right bottom) in Wisconsin, 1999 [8] (Photo Credit: Space Science and Engineering Center at University of Wisconsin-Madison and WisconsinView); (b) A prototype of autonomous robotic fish developed by the Smart Microsystems Laboratory at Michigan State University [7].

in capturing large-scale spatially distributed phenomena of interest. An alternative approach is *in-situ* sensing with fixed or buoyed/moored sensors [4]. However, since buoyed sensors cannot move around, they have limited adaptability in monitoring dynamic aquatic processes like HABs. With advances in underwater robotics and wireless networking, there is a growing interest in using underwater sensor platforms like autonomous underwater vehicles (AUVs) [5] and sea gliders [6] to monitor the environment. However, it is difficult to deploy many AUVs or sea gliders due to their high manufacturing and operational costs.

In this paper, we propose to use inexpensive, low-power robotic sensor platforms to sample and reconstruct spatiotemporal aquatic processes of interest. Fig. 1(b) shows a prototype of such platforms called *robotic fish*. Each robotic fish is equipped with onboard batteries, ZigBee wireless interface, control, localization and navigation modules [7], and can be interfaced with various aquatic sensors. Robotic fish can form an autonomous network and sense aquatic environment at fine spatial and temporal granularities.

Aquatic sensor networks composed of robotic fish are a typical Cyber-Physical System (CPS) whose efficient operation depends on the tight coupling and coordination between cyber (sensing, communication, and information processing) and physical components (mobility control and environment). Compared with terrestrial sensor networks, there are several unique challenges associated with aquatic sensor networks, including uncontrollable disturbances from the underlying fluid medium (e.g., waves and flows), inherently

dynamic profiles of aquatic processes, and significant errors in motion control. Therefore, both sensing and mobility control of robotic fish must account for the spatial variability and temporal evolution of aquatic processes. Moreover, our measurements show that aquatic sensors equipped with ZigBee radio have highly variable link quality and only about half of the communication range of the terrestrial radio. Such characteristics must be explicitly considered in the design of the network. Finally, the operation of these sensors has to be very energy-efficient due to the limited power supply.

We make the following key contributions in this paper:

- 1) We propose a new approach to the sampling and reconstruction of spatiotemporal aquatic field using a sensor *swarm* composed of inexpensive, low-power, and collaborative robotic sensors. Our approach features a *rendezvous-based* mobility control scheme, where sensors in a swarm gather and sense the environment in a series of carefully chosen rendezvous regions, reducing the overhead of inter-sensor coordination during movement.
- 2) We design a novel feedback control algorithm that maintains the desirable level of wireless connectivity of a sensor swarm in the presence of significant physical dynamics. Based on a wireless signal propagation model, the control-theoretic algorithm adjusts the radius of rendezvous region adaptively to ensure a bound on the packet reception ratio (PRR) between sensors.
- 3) We present a new analysis of spatiotemporal field reconstruction accuracy based on mutual information and posterior entropy. Our analytical results are used to guide the selection of rendezvous regions so that the reconstruction accuracy can be maximized subject to the limited sensor mobility.
- 4) We evaluate our approach through extensive simulations based on real data traces of water surface temperature field and on-water ZigBee wireless communication. The results show that a sensor swarm can maintain desirable network connectivity level and accurately reconstruct large, dynamic aquatic fields. Moreover, our implementation on sensor hardware provides important insights into the feasibility of adopting advanced information-theoretic movement scheduling algorithms on low-power robotic sensor platforms.

The rest of this paper is organized as follows. Section II reviews related work. Section III introduces the background and provides an overview of our approach. Section IV presents the control-theoretic connectivity maintenance algorithm. Section V presents the information-theoretic swarm movement scheduling algorithms. Section VI presents the results of extensive trace-driven simulations and implementation on sensor platform. Section VII concludes this paper.

## II. RELATED WORK

Sampling and reconstruction of physical field using networked sensor systems has recently received increasing interest. Early work focuses on stationary sensor deployment. In [9], positions of sensors are selected before the real deployment to reduce the uncertainty in reconstructing a spatial physical field that follows the Gaussian process. However, the proposed algorithms are computationally intensive and hence can only be executed offline. Recently, mobility has been exploited to enhance the adaptability and sensing capability of sensor systems. In [10], a robotic boat supplements a static sensor network to reduce the field reconstruction error, where the boat's movement is guided by the measurements of the sensor network. A recent study [11] develops active learning schemes for mobile sensor networks, which plan the movements of mobile sensors based on the feedback of previous measurements. Several recent studies focus on leveraging sensors' mobility to reconstruct physical fields that follow the Gaussian process. In [2], the movement of mobile sensors is directed to reduce the uncertainties in estimating the field variables at a set of pre-specified locations. The algorithms developed for placing stationary sensors in [9] are extended to schedule the movement of a sensor network in reconstructing a Gaussian process [12]. However, these studies do not account for the constraints of low-power robotic sensor systems, such as the limited motion, computation and communication capabilities. Moreover, the existing studies focus on open-loop solutions that often fail to adapt to highly complex and dynamic aquatic environment. In contrast, this paper aims to develop practical and adaptive wireless communication, sensing, and movement control approaches for mobile CPSs in aquatic field reconstruction.

Most previous works on maintaining sensor network connectivity adopt the graph theory [2] and potential field theory [13], and assume fixed communication range and reliable communication quality. Recent studies have revealed significant stochasticity and irregularity in link quality of low-power wireless sensors [14]. In this paper, we aim to adaptively maintain the network connectivity defined based on the average PRR of a robotic sensor swarm in the presence of environment and system dynamics. Feedback control has been widely adopted to improve the adaptability of computing systems [15]. Different from existing solutions, our control-theoretic connectivity maintenance algorithm specifically deals with the dynamics caused by movement of robotic sensor swarm and disturbances from the aquatic environment.

## III. OVERVIEW OF APPROACH

### A. Background and Challenges

Our objective is to reconstruct an aquatic scalar field that follows the spatiotemporal Gaussian process using a group

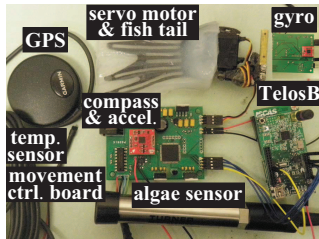


Figure 2. Internal components of the robotic fish [7].

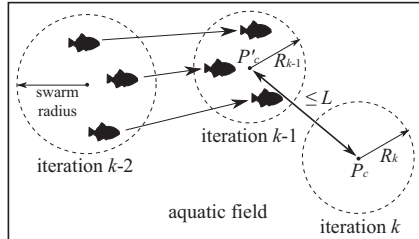


Figure 3. Rendezvous-based swarm scheme. Dashed circles represent the rendezvous circles.

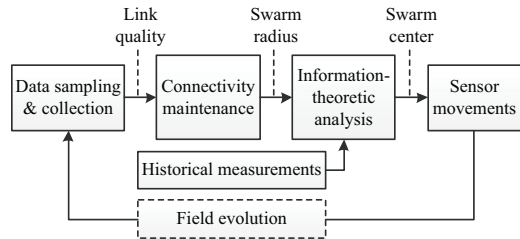


Figure 4. The iterative sampling process of a robotic sensor swarm.

of robotic sensors. Different from existing solutions, our approach is based on inexpensive robotic sensor platforms exemplified by the robotic fish developed in our previous work [7], as shown in Fig. 1(b). Fig. 2 shows the internal components of the robotic fish, which include a TelosB mote for communication and data processing, a movement control board, a GPS module, and various aquatic sensors. Due to the resource constraints, robotic fish has limited capabilities of computation, communication, and movement. For instance, the TelosB mote only has an 8MHz MCU and a low-power 802.15.4 radio with short communication range. In this paper, we aim to develop a practical approach for aquatic field reconstruction, which addresses the complex uncertainties/dynamics of the monitored physical field and the constraints of realistic robotic sensor platforms.

The design of our approach is motivated by the following major challenges in reconstructing a spatiotemporal field. First, the physical and biological phenomena of interest often affect large spatial areas. For instance, HABs can spread over the water area of a dozen to tens of square kilometers (e.g., Lake Monona and Lake Mendota, Wisconsin, shown in Fig. 1(a) [8]). However, the number of robotic sensors available in practice is often small (e.g., a few dozens). Moreover, as the robotic sensors in aquatic environment often have short communication ranges, the area that the robotic sensor system can sample at any given time is limited. Second, because of the complex environment dynamics (e.g., wave and wind) and the limited motion capabilities of the robotic sensors, accurate movement control of an aquatic sensor system is often challenging. Third, the link quality and network connectivity of robotic sensors are highly dynamic due to physical uncertainties. The resulted data loss can significantly affect the accuracy of field reconstruction.

### B. System Model

To address the aforementioned challenges, we adopt a novel *rendezvous-based swarm* scheme as illustrated in Fig. 3. We assume that all sensors know their positions and are time-synchronized, e.g., through GPS or in-network localization/synchronization services. The robotic sensor system iteratively samples the aquatic field. In each *sampling iteration*, the sensors move into a *rendezvous circle*, form a

*swarm* and sample the environment. In the swarm, a sensor serves as the *swarm head*, which collects the measurements of other sensors via wireless communications and computes the location of the rendezvous circle for the next iteration. To simplify the data collection process and reduce communication overhead, the swarm adopts a single-hop star network topology centered at the swarm head. Moreover, to balance the energy consumption of sensors, the swarm head role can rotate among all sensors. Such a swarm scheme allows the robotic sensor system to efficiently collaborate in sensing a large dynamic aquatic field.

To address the challenge of limited mobility, the sensors are scheduled to move to randomly selected positions in the next rendezvous circle. Small motion control errors can be tolerated as long as the final positions of sensors fall within the rendezvous circle. Moreover, the communication overhead is low because sensors coordinate with each other only when they gather in a rendezvous circle. Therefore, this movement scheme is practical for low-power aquatic robotic platforms [7] [10].

### C. Approach Overview

We now present an overview of our cyber-physical approach for sampling the aquatic field using robotic sensor swarm. Initially, the swarm is dropped at a venue within the region affected by the physical/biological process of interest. As shown in Fig. 4, in each sampling iteration, all sensors take measurements and send to the swarm head. The swarm head assesses the quality of the network connectivity based on the received data and then determines the radius of the rendezvous circle (referred to as *swarm radius*) in the next sampling iteration. Given the projected swarm radius, the swarm head then conducts information-theoretic analysis to select the location of the next rendezvous circle to maximize the improvement of the field reconstruction accuracy. After that, the swarm head generates random target positions within the next rendezvous circle and assigns the positions to each sensor to minimize the total moving distance. The target positions are finally sent to the sensors for directing their movements. Our approach has the following two key novelties.

**Control-theoretic connectivity maintenance:** Data loss of

wireless communication can significantly affect the quality of sensing. A key goal of our system is to ensure that the swarm head reliably receives the measurements from all sensors. However, this is challenging because the on-water wireless links have highly dynamic quality due to the impact of fluid medium and changing positions of sensors during movement. We develop a control-theoretic algorithm to maintain desirable connectivity of a sensor swarm in the presence of these dynamics by adaptively adjusting the swarm radius. Specifically, the swarm head first estimates the quality of network connectivity based on the average of PRRs of all links. As the swarm average PRR generally decreases with the swarm radius, the swarm head calculates a new swarm radius based on a wireless signal propagation model and the current swarm average PRR, such that the expected connectivity in the next sampling iteration can be maintained at a desirable level. A control problem is formulated to address this problem and its solution gives an adaptive algorithm for tuning the swarm radius.

**Information-theoretic movement scheduling:** Due to limited power supply and high power consumption in locomotion, the sensor swarm must efficiently schedule the movement of sensors to sample the field. Specifically, the swarm head must find the location of the next rendezvous circle subject to energy budget, such that the improvement of the field reconstruction accuracy can be maximized with the newly obtained sensor measurements. In this paper, we employ information-theoretic analysis to guide the selection of rendezvous circle locations. Moreover, two information metrics (i.e., mutual information and posterior entropy) with different computational complexities can be integrated with our analysis, which hence allow the system designer to choose desirable trade-offs between the system overhead and reconstruction accuracy.

#### IV. SWARM CONNECTIVITY MAINTENANCE

A sensor swarm must form a connected network to be able to collaborate in the field sampling and coordinate each other's movement. However, the wireless connectivity between a robotic sensor and the swarm head is affected by environment and system dynamics, which include the stochastic fluctuation of the on-water wireless links, the errors of localization and motion control, and the uncertain distance between moving sensors. Fig. 5 plots the PRR traces of on-water communication between two IRIS motes measured in Lake Lansing, Michigan. It can be seen that the PRR shows significant variance, which is mostly caused by the radio and environment dynamics [14]. Such highly dynamic communication quality can lead to increased communication cost in the field sampling and even loss of sensors due to disconnected network. To adapt to these dynamics, we formulate the swarm connectivity maintenance as a feedback control problem, which aims to maintain the

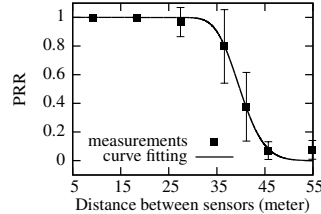


Figure 5. The PRR measurements versus the distance between two sensors. The error bar represents the standard deviation.

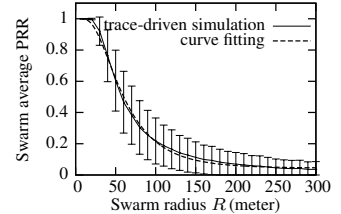


Figure 6. The swarm average PRR versus the swarm radius. The error bar represents the standard deviation.

swarm connectivity at a desired level by adjusting the swarm radius based on the quality of all links measured at run time.

##### A. Modeling Swarm Connectivity

As discussed in Section III-B, the sensor swarm forms a network with single-hop star topology in a rendezvous circle. Compared with multi-hop topology, the single-hop topology of sensor swarm incurs significantly lower overhead in communication and network formation/maintenance. As shown in Fig. 5, the reliable on-water communication range of a typical 802.15.4 radio is about 35 m. As a result, a sensor swarm can spread over an area of up to 3,800 m<sup>2</sup>. We adopt the average PRR of the links between the swarm head and all sensors as the metric of swarm connectivity. This metric quantifies not only the average connectivity of the swarm but also the communication cost in collecting sensor measurements in a sampling iteration. This section derives the expression for the average PRR given swarm radius, which allows us to adaptively control the swarm connectivity by adjusting the swarm radius.

Let  $P_t$  (in dBm) denote the power of the wireless signal transmitted by a sensor, and  $PL(d_0)$  (in dBm) denote the path loss at reference distance  $d_0$ . The signal power at the receiver that is  $d$  meters from the transmitter is  $P_r(d) = P_t - PL(d_0) - 10\alpha \log_{10}(d/d_0)$  [16], where  $\alpha$  is the path loss exponent that typically ranges from 2.0 to 4.0. We assume that the noise power (denoted by  $P_n$ ) in dBm follows the zero-mean normal distribution with variance  $\xi^2$  [16]. The signal-to-noise ratio (SNR) at distance  $d$  is given by  $\text{SNR} = P_r(d) - P_n$ . We assume that a packet can be successfully received if the SNR is greater than a threshold denoted by  $\eta$  [17]. Hence, the PRR of a single link can be derived as

$$\text{PRR}(d) = 1/2 + 1/2 \cdot \text{erf}(a_1 \log_{10} d + a_2), \quad (1)$$

where  $a_1 = -5\sqrt{2}$ ,  $a_2 = \frac{P_t - PL(d_0) - \eta}{\sqrt{2}\xi} + 5\sqrt{2} \log_{10} d_0$ , and  $\text{erf}(\cdot)$  is the error function. We now use the real PRR traces of on-water 802.15.4 wireless link to verify the above model. Fig. 5 plots the PRR measured by two IRIS motes versus distance in an experiment conducted on the wavy water surface of Lake Lansing, Michigan, on a windy day. Specifically, we placed the two motes about 12 cm above the water surface and measured the PRR versus the distance between

the two motes. Each PRR measurement was calculated from 50 packets transmitted within one second. According to our experience, the communication range of IRIS mote on water surface decreases by about 50% compared to that on land. Moreover, the wireless link in such a wavy water environment is more dynamic than that in calm water environment, due to the multipathing effect. The least square fitting of the average of the PRR measurements versus distance is  $1/2 + 1/2 \cdot \text{erf}(-7.096 \log_{10} d + 26.14)$ , which is also plotted in Fig. 5. We can see that the fitted value for  $a_1$  (i.e.,  $-7.096$ ) is very close to its theoretical value (i.e.,  $-5\sqrt{2} = -7.0711$ ). Moreover, the fitted curve well matches the average of the PRR measurements. Therefore, the model in Eq. (1) can characterize the average performance of on-water link PRR. From Fig. 5, we also observe that the PRR measurements exhibit significant variance especially in the transition range from 25 m to 40 m. Although Eq. (1) only captures the expected PRR, the control-theoretic connectivity maintenance algorithm presented in Section IV-B accounts for the variance of PRR measurements.

Based on the above single-link PRR model, we now derive the average PRR over all sensors that are randomly distributed within the rendezvous circle. Our analysis shows that it is difficult to derive the closed-form formula for the average PRR. We propose an approximate formula as follows. The expectation of the distance between any sensor and the swarm head (denoted by  $\mathbb{E}[d]$ ), which are two random points in the rendezvous circle, is a linear function of the swarm radius (denoted by  $R$ ), specifically,  $\mathbb{E}[d] = 128R/45\pi$ . Based on this observation and Eq. (1), we approximate the average PRR over all sensors (denoted by  $\overline{\text{PRR}}(R)$ ) by

$$\overline{\text{PRR}}(R) \simeq (1 - c) + c \cdot \text{erf}(c_1 \log_{10} R + c_2), \quad (2)$$

where  $c_1$  ( $c_1 < 0$ ),  $c_2$  ( $c_2 > 0$ ), and  $c$  ( $0 < c < 0.5$ ) are three coefficients. Although Eq. (2) is an approximate model, the feedback-based connectivity maintenance algorithm can tolerate minor inaccuracy in system modeling. We conduct Monte Carlo simulations to verify the accuracy of the above approximate model, and determine the values of the three coefficients. Specifically, for a given  $R$ , we generate a large number (20,000) of random placements of 10 sensors in the rendezvous circle. In the simulations, the PRR of each link is set to be the distance-based interpolation of real PRR measurements obtained in the aforementioned on-water experiment. Fig. 6 shows the error bar of the swarm average PRR, where the variances are caused by the random sensor placements and estimation inaccuracy as well as the inherent stochasticity of wireless link. We then fit the curve defined by Eq. (2) with the simulation results, as shown in Fig. 6. From the figure, we can see that the approximate model for the swarm average PRR is fairly accurate. The fitted value for the coefficient  $c_1$ ,  $c_2$ , and  $c$  are  $-1.201$ ,  $4.879$ , and  $0.4783$ , respectively. These values are also adopted in the

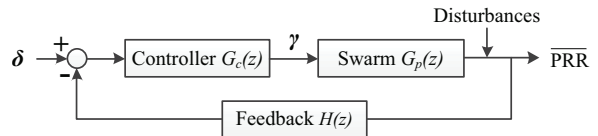


Figure 7. The closed loop for connectivity control.

performance evaluation in Section VI.

### B. Swarm Connectivity Control

Our objective is to maintain the swarm average PRR at a desired level (denoted by  $\delta$ ) in the presence of various environment and system dynamics. From Fig. 6, the swarm average PRR decreases with the swarm radius. However, the amount of information sampled by the sensors often increases with the swarm radius. Therefore, there is a trade-off between the amount of information obtained by the swarm and its connectivity. To avoid the loss of sensors that can have catastrophic consequence to the swarm, we ensure that the swarm is a well connected network in each rendezvous circle by setting a relatively high  $\delta$ , e.g., 0.8 to 0.9. In this section, we first analyze the control laws based on the connectivity model in Eq. (2) and then develop the connectivity maintenance algorithm.

The block diagram of the feedback control loop is shown in Fig. 7, where  $G_c(z)$ ,  $G_p(z)$  and  $H(z)$  represent the transfer functions of the connectivity maintenance algorithm, the sensor swarm system and the feedback. Specifically, the desired PRR level  $\delta$  is the *reference*, and the  $\overline{\text{PRR}}(R)$  is the *controlled variable*. As  $\overline{\text{PRR}}(R)$  is a nonlinear function of  $R$  (c.f. Eq. (2)), we define  $\gamma = \text{erf}(c_1 \log_{10} R + c_2)$  as the *control input* to simplify the controller design. As a result,  $\overline{\text{PRR}}(\gamma) \simeq (1 - c) + c \cdot \gamma$ , and its  $z$ -transform is  $G_p(z) = c$ . In each sampling iteration, to ensure that the swarm head receives the measurements from all sensors, a sensor retransmits the lost packet until it receives an acknowledgement from the swarm head. At the end of each sampling iteration, the swarm head estimates the  $\overline{\text{PRR}}(R)$  as  $\frac{1}{N} \sum_{i=1}^N \frac{1}{\text{CTX}_i}$ , where  $N$  is the number of sensors in the sensor swarm, and  $\text{CTX}_i$  is the number of (re-)transmissions of sensor  $i$  in the current sampling iteration. Such a passive estimation approach avoids transmitting a large number of measurement packets for estimating PRRs. Then, the swarm head updates  $\gamma$  based on the estimated  $\overline{\text{PRR}}(R)$ , and sets  $R$  in the next sampling iteration accordingly. As the feedback will take effect in the next iteration,  $H(z) = z^{-1}$ , which represents a delay of one iteration. Since the system is of zero order, a first-order controller is sufficient to achieve the stability and convergence [18]. Hence, we let  $G_c(z) = \frac{\alpha}{1 - \beta \cdot z^{-1}}$ , where  $\alpha > 0$  and  $\beta > 0$ . The settings of  $\alpha$  and  $\beta$  need to ensure the system stability, convergence and robustness. Following the standard method for analyzing stability and convergence [18], the stability and convergence condition can be obtained as  $\beta = 1$  and  $0 < \alpha < 2/c$ .

In this paper, we model three uncertainties that substantially affect the  $\overline{\text{PRR}}(R)$  as the disturbances in the control loop shown in Fig. 7. First, as shown in Fig. 5, the PRR measurements exhibit variance especially in the transition range from 25 m to 40 m. Second, the swarm topology changes with the random sensor positions, hence also causes variance to the  $\overline{\text{PRR}}(R)$ . Third, although the estimated PRR from the number of (re-)transmissions is unbiased, it has variance because of the limited number of samples. The error bars in Fig. 6 show the overall standard deviation versus the swarm radius. From the figure, we find that in order to keep a satisfactory swarm average PRR around 0.8, the standard deviation is 0.12. We now discuss how to design  $G_c(z)$  to reduce the impact of such random disturbances. From control theory [18], to minimize the effects of disturbance on the controlled variable  $\overline{\text{PRR}}(R)$ , the gain of  $G_c(z)G_p(z)H(z)$  should be made as large as possible. By jointly considering the stability and convergence condition, we set  $\alpha = 2b/c$  where  $b$  is a relatively large value within  $[0, 1]$ . In the experiments conducted in this paper,  $b$  is set to be 0.9.

Implementing  $G_c(z)$  in the time domain gives the connectivity maintenance algorithm. According to Fig. 7, we have  $G_c(z) = \gamma(z)/(\delta - H(z)\overline{\text{PRR}})$ . From  $H(z)$  and  $G_c(z)$ , the control input can be expressed as  $\gamma(z) = z^{-1}\gamma(z) + 2bc^{-1}(\delta - z^{-1}\overline{\text{PRR}})$ , and its time-domain implementation is  $\gamma_k = \gamma_{k-1} + 2bc^{-1}(\delta - \overline{\text{PRR}}_{k-1})$ , where  $k$  is the index of sampling iteration. The swarm radius to be set in the  $k^{\text{th}}$  sampling iteration is given by  $R_k = 10^{(\text{erf}^{-1}(\gamma_k) - c_2)/c_1}$ .

## V. INFORMATION-THEORETIC MOVEMENT SCHEDULING

In this section, we first briefly introduce the Gaussian process model that characterizes many physical/biological phenomena, and present the field reconstruction algorithm. We then present the information-theoretic analysis for selecting the location of rendezvous circle in the next sampling iteration, which aims to maximize the accuracy of field reconstruction.

### A. Physical Field Model and Reconstruction Algorithm

We assume that the monitored physical phenomenon follows the spatiotemporal Gaussian process. Let  $Z(p, t)$  denote the field variable at point  $p \in \mathbb{R}^2$  and time  $t \in [0, +\infty]$ . For instance, the surface phytoplankton population density is an important field variable of HABs. A Gaussian process can be fully characterized by the *mean function*, denoted by  $\mathcal{M}(p, t)$ , and the *covariance function*, denoted by  $\mathcal{K}((p, t), (p', t'))$ , where  $(p, t)$  and  $(p', t')$  are two time-space coordinates. In this paper, we adopt the following covariance function that has been widely adopted [1] [2] [9]:  $\mathcal{K}(d, \Delta t) = \sigma^2 \cdot \exp(-d^2/(2\zeta_s^2)) \cdot \exp(-\Delta t^2/(2\zeta_t^2))$ , where  $d = \|p - p'\|_{\ell_2}$ ,  $\Delta t = |t - t'|$ ,  $\sigma^2$  is the prior variance of any field variable,  $\zeta_s$  and  $\zeta_t$  are the spatial and temporal *kernel bandwidths*. Therefore, the covariance

function can be rewritten as  $\mathcal{K}(d, \Delta t)$ . The vector composed of the field variables at  $N$  time-space coordinates  $\{(p_i, t_i) \mid i \in [1, N]\}$ , denoted by  $\mathbf{Z}$ , follows the multivariate Gaussian distribution, i.e.,  $\mathbf{Z} \sim \mathcal{N}(\mathbf{m}, \Sigma)$ , where  $\mathbf{m}$  and  $\Sigma$  are the mean vector and covariance matrix. Specifically,  $\mathbf{m} = [\mathcal{M}(p_1, t_1), \dots, \mathcal{M}(p_N, t_N)]$  and the  $(i, j)^{\text{th}}$  entry of  $\Sigma$  is given by  $\mathcal{K}(\|p_i - p_j\|_{\ell_2}, |t_i - t_j|)$ . Sensor measurements can be corrupted by noises from the sensor circuitry and environment [2]. The reading at time-space coordinates  $(p, t)$ , denoted by  $R(p, t)$ , is given by  $R(p, t) = Z(p, t) + W$ , where  $W$  is a zero-mean Gaussian noise with variance of  $\sigma_w^2$ .

We now discuss how to reconstruct the field using all measurements. To facilitate the expression, we define  $\mathbf{H}$  as a row vector composed of all measurements, i.e.,  $\mathbf{H} = [R(p_1, t_1), \dots, R(p_N, t_N)]$ ,  $\mathbf{m}$  as a row vector composed of the corresponding prior mean values, and  $T$  as the time duration of each sampling iteration. Therefore, each  $t_i$  ( $i \in [1, N]$ ) is always multiple of  $T$ . The  $\mathbf{H}_c$  is a  $3 \times N$  matrix, where each column is the time-space coordinates of the corresponding measurement in  $\mathbf{H}$ . The objective of reconstructing a Gaussian process field is to estimate the posterior mean and variance at any time-space coordinates  $(p, t)$  given  $\mathbf{H}$ , which are given by

$$\mathbb{E}[Z|\mathbf{H}] = \mathcal{M}(p, t) + \tilde{\Sigma}[(p, t), \mathbf{H}_c] \cdot \tilde{\Sigma}^{-1}[\mathbf{H}_c] \cdot (\mathbf{H} - \mathbf{m})^T, \quad (3)$$

$$\text{Var}[Z|\mathbf{H}] = \sigma^2 - \tilde{\Sigma}[(p, t), \mathbf{H}_c] \cdot \tilde{\Sigma}^{-1}[\mathbf{H}_c] \cdot \tilde{\Sigma}^T[(p, t), \mathbf{H}_c], \quad (4)$$

where  $\mathbb{E}[\cdot]$  and  $\text{Var}[\cdot]$  denote expectation and variance,  $\tilde{\Sigma}$  is a matrix calculated from the covariance matrix  $\Sigma$  of the field variables at  $\mathbf{H}_c$ . Specifically, the  $(i, j)^{\text{th}}$  entry of  $\tilde{\Sigma}$  is given by  $\tilde{\Sigma}_{ij} = \Sigma_{ij} + \theta_{ij} \frac{\sigma_w^2}{\sigma^2}$ , where  $\theta_{ij} = 1$  if  $i = j$ , and otherwise  $\theta_{ij} = 0$ . There are three interesting observations from Eqs. (3) and (4). First, because of the spatiotemporal correlation, the posterior variance (i.e., the uncertainty) is reduced given the measurements  $\mathbf{H}$ . Second, from Eq. (4), the posterior variance does not depend on the prior and posterior means. As our movement scheduling algorithm aims to reduce the variance, it does not need the knowledge of means. Third, as  $\tilde{\Sigma}$  often has a high dimension due to a large number of measurements, it is infeasible to compute its inversion in Eqs. (3) and (4) on resource-constrained robotic sensors. Therefore, the reconstruction algorithm needs to be executed after all historical sensor measurements are fetched back to a data processing center.

### B. Information-Theoretic Swarm Center Selection

We now discuss the selection of the center of the next rendezvous circle (referred to as *swarm center*), which aims to improve the accuracy of the field reconstruction algorithm (i.e., Eqs. (3) and (4)). Suppose that the swarm has  $N$  robotic sensors and will schedule the sensor movements for the next (i.e., the  $k^{\text{th}}$ ) sampling iteration. Let  $\mathbf{V}$  denote the region to be reconstructed, and  $\mathbf{S}$  denote the set of

target time-space coordinates for all sensors.<sup>1</sup> Hence,  $\mathbf{S}$  can be represented as  $(\{p_1, p_2, \dots, p_N\}, kT)$ , where  $p_i$  is the target position of sensor  $i$ . Let  $p'_c$  and  $p_c$  denote the swarm center in the  $(k-1)$ <sup>th</sup> and  $k$ <sup>th</sup> sampling iteration, and  $R_k$  is the scheduled swarm radius for the  $k$ <sup>th</sup> iteration by the connectivity maintenance algorithm. The optimal solution of  $\mathbf{S}$  maximizes the following information-theoretic metric:

$$\Omega(\mathbf{S}) = \mathbb{H}[\mathbf{V} \setminus \mathbf{S} \mid \mathbf{H}_c] - \mathbb{H}[\mathbf{V} \setminus \mathbf{S} \mid \mathbf{H}_c \cup \mathbf{S}], \quad (5)$$

subject to

$$\|p'_c - p_c\|_{\ell_2} \leq L; \quad \|p_i - p_c\|_{\ell_2} \leq R_k, \forall i \in [1, N], \quad (6)$$

where the  $\mathbb{H}[\cdot]$  denotes entropy and quantifies the uncertainty. The above problem aims to maximize the drop of uncertainty at all the ungauged sites by sampling the field variables at  $\mathbf{S}$  given the historical measurements at  $\mathbf{H}_c$ . The constraint in the first part of Eq. (6) specifies the reachable area of the swarm due to limited sensor movement speed. For instance, we can set  $L = v \cdot T$ , where  $v$  is the maximum speed of the robotic sensors. The constraint in the second part of Eq. (6) ensures the scheduled swarm radius. These constraints are also illustrated in Fig. 3. A similar problem without the condition  $\mathbf{H}_c$  and the constraints in Eq. (6) has been proven to be NP-hard [19]. Hence, the above problem has prohibitively high complexity that is not practical for robotic sensor platforms. In this paper, we propose a heuristic approach to approximate the whole swarm by the swarm center, which can largely reduce the computation overhead. In this heuristic approach, we adopt *mutual information* (MI) and *posterior entropy* (PE) to quantify the information reward. As these two metrics differ in computation overhead and the resulted reconstruction accuracy, they allow the system designer to choose desirable trade-off between the overhead and accuracy subject to the budgets of computation resources of robotic sensors.

We first discuss the MI-based metric. The MI of a random variable  $X$  given a set of random variables  $\mathbf{Y}$  can be expressed as  $\mathbb{I}[X; \mathbf{Y}] = \mathbb{H}[X] - \mathbb{H}[X \mid \mathbf{Y}]$ , where  $\mathbb{H}[X \mid \mathbf{Y}] = \frac{1}{2} \log(2\pi e \cdot \text{Var}[X \mid \mathbf{Y}])$  and

$$\text{Var}[X \mid \mathbf{Y}] = \text{Var}[X] - \Sigma[X, \mathbf{Y}] \cdot \Sigma^{-1}[\mathbf{Y}] \cdot \Sigma^T[X, \mathbf{Y}]. \quad (7)$$

The  $\Sigma[X, \mathbf{Y}]$  is a row vector composed of the covariances of  $X$  with each variable in  $\mathbf{Y}$ , and  $\Sigma^{-1}[\mathbf{Y}]$  is the inverse of the covariance matrix of  $\mathbf{Y}$ . Given available measurements at  $\mathbf{H}_c$ , the MI-based information reward for the swarm centered at position  $p_c$ , denoted by  $\Omega_{\text{MI}}(p_c, kT)$ , is defined as

$$\begin{aligned} \Omega_{\text{MI}}(p_c, kT) &= \mathbb{I}[\mathbf{V} \setminus (p_c, kT); (p_c, kT) \mid \mathbf{H}_c] \\ &= \mathbb{H}[(p_c, kT) \mid \mathbf{H}_c] - \mathbb{H}[(p_c, kT) \mid \mathbf{V} \cup \mathbf{H}_c \setminus (p_c, kT)]. \end{aligned}$$

<sup>1</sup>To simplify the presentation,  $\mathbf{V}$  refers to both the set of field variables and the corresponding time-space coordinates. So do  $\mathbf{S}$  and  $\mathbf{H}_c$ .

The above information reward metric indicates the drop of uncertainty about the region outside the swarm given all historical measurements if the swarm is centered at  $p_c$  in the next iteration. The swarm center selection is hence to maximize  $\Omega_{\text{MI}}$ , subject to the constraints in Eq. (6).

The complexity for computing  $\Omega_{\text{MI}}$  for a certain  $p_c$  is  $O(|\mathbf{V}|^2)$ . However, as the aquatic phenomenon of interest (e.g., HABs) often affects a large spatial area, computing  $\Omega_{\text{MI}}$  can incur high overhead. To reduce the computation overhead, we propose another information reward metric based on PE:

$$\Omega_{\text{PE}}(p_c, kT) = \mathbb{H}[(p_c, kT) \mid \mathbf{H}_c].$$

Different from  $\Omega_{\text{MI}}$ ,  $\Omega_{\text{PE}}$  indicates the uncertainty about the rendezvous circle centered at  $p_c$  in the next iteration given the historical measurements. For each certain  $p_c$ , the complexity of computing  $\Omega_{\text{PE}}$  is  $O(|\mathbf{H}_c|^2)$ , which is much smaller than that of  $\Omega_{\text{MI}}$ . Although such a metric does not necessarily lead to the maximum uncertainty drop for the ungauged sites, it can reduce the computation overhead by only considering the most uncertain positions.

As discussed in Section III, once the swarm center and radius are determined, the swarm head randomly selects  $N$  positions in the rendezvous circle. We find the one-to-one mapping from the current positions of sensors to the newly selected positions, such that the sum of sensors' movement distances is minimized. This can be solved by Munkres assignment algorithm with a complexity of  $O(N^3)$ . The swarm head finally sends the target position to each robotic sensor, which then moves toward the target position.

### C. Truncating Historical Measurements

Both the metrics  $\Omega_{\text{MI}}$  and  $\Omega_{\text{PE}}$  involve storing and inverting the covariance matrix  $\Sigma[\mathbf{H}_c]$  when computing Eq. (7). This imposes substantial challenges to the robotic sensor platforms with limited computation resources. For instance, a TelosB mote equipped with 10 KB RAM can store at most a  $50 \times 50$  covariance matrix. Moreover, matrix inversion is a computation-intensive operation with at least cubic complexity with respect to the number of historical measurements. To develop practical information-theoretic movement scheduling algorithms for robotic sensors, we propose two schemes for truncating the historical measurements. Both schemes select  $K$  measurements to compose the covariance matrix.

The first scheme selects  $K$  historical measurements with the largest covariances regarding a candidate swarm center. This scheme is referred to as *cov-trunc*. The rationale of *cov-trunc* is as follows. As we only use a subset of historical measurements, the conditional variance in Eq. (7) will increase. The *cov-trunc* scheme maximizes each element in  $\Sigma[X, \mathbf{Y}]$ , and hence can efficiently suppress the undesired increase of the conditional variance caused by the truncation. The drawback of *cov-trunc* is that it needs to truncate the

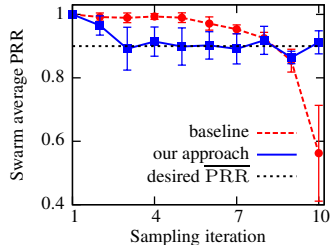


Figure 8. Swarm average PRR versus sampling iteration. The error bar represents the standard deviation.

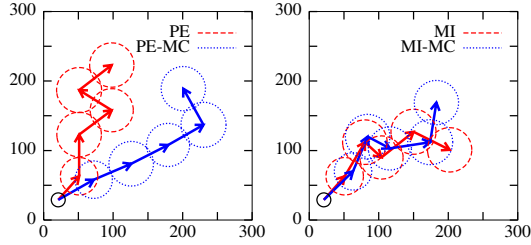


Figure 9. Trajectories of a robotic sensor swarm with 10 sensors in the first 6 sampling iterations in the reconstruction of a  $300 \times 300 \text{ m}^2$  field.

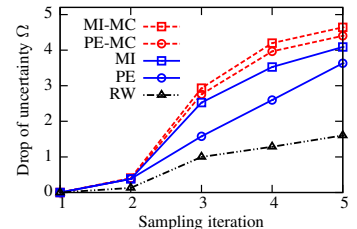


Figure 10. Information reward versus sampling iteration.

historical measurements for each candidate swarm center when maximizing  $\Omega_{\text{MI}}$  and  $\Omega_{\text{PE}}$ . As a result, a matrix inversion operation is needed for each candidate swarm center, which results in high computation overhead for the swarm head. To address this issue, we propose another truncating scheme, referred to as *time-trunc*. The *time-trunc* selects the most recent  $K$  historical measurements. As the most recent sampling positions are generally in the proximity of the swarm in the next iteration, *time-trunc* can well approximate *cov-trunc* even though it ignores the spatial correlation. The *time-trunc* scheme has the following two advantages. First, it only needs a matrix inversion operation for each sampling iteration. Second, the swarm head only needs to maintain a first-in-first-out historical measurement buffer with size of  $K$ . This buffer can be easily migrated in the swarm head rotation process for the purpose of balancing energy consumption.

## VI. PERFORMANCE EVALUATION

We evaluate the performance of the proposed algorithms by trace-driven simulations and implementation on hardware. First, we evaluate the connectivity maintenance and the swarm movement scheduling algorithms using extensive simulations based on real data traces of water surface temperature field [20] and on-water ZigBee wireless communication. Second, we implement one of the proposed swarm movement scheduling algorithms on TelosB sensor platform and evaluate its overhead. The results provide insight into the feasibility of adopting advanced information-theoretic movement scheduling algorithms on mote-class robotic sensor platforms.

### A. Trace-Driven Simulations

1) *Simulation Methodology and Settings*: In the simulations, 10 robotic sensors are used to reconstruct a scalar field in a square region. The hyperparameters of the Gaussian process are set to be  $[\sigma^2, \varsigma_s, \varsigma_t] = [9, 6, 8]$ , unless otherwise specified. Note that these settings are consistent with [10] [12] and obtained from real on-water temperature traces [20]. Initially, the robotic sensors are randomly deployed in a small region with radius of 10 m. In each sampling iteration, the PRR of each link is set to be the distance-based

interpolation of real on-water PRR traces measured by two IRIS motes on Lake Lansing, Michigan (c.f. Section IV-A). Other settings include: desired swarm connectivity level  $\delta = 0.9$ , sampling iteration duration  $T = 5$  min, sensor movement speed  $v = 0.2$  m/s, and  $L = v \times T = 60$  m.

2) *Swarm Connectivity Maintenance*: We first compare our connectivity maintenance algorithm with a heuristic baseline algorithm. The heuristic algorithm adopts Kalman filter to update the coefficient  $c$  in Eq. (2) based on the recently estimated  $\overline{\text{PRR}}(R)$ . The next swarm radius is then obtained by solving Eq. (2). Fig. 8 plots the  $\overline{\text{PRR}}(R)$  in the first 10 sampling iterations. The error bars, calculated from 20 runs, are caused by the disturbances discussed in Section IV-B. We can see that the swarm average PRR controlled by our algorithm quickly converges to the desired connectivity level. The range of swarm radius after 10 iterations is [24, 36]. In contrast, as the heuristic algorithm does not tune the swarm radius directly based on the estimated swarm average PRR, it does not converge as shown in Fig. 8.

3) *Effectiveness of Swarm Center Selection*: We now compare the two swarm center selection approaches presented in Section V-B (referred to as MI and PE) with three other baseline approaches. The first baseline (referred to as MI-MC) finds the next swarm center according to the metric  $\Omega(\mathbf{S})$  in Eq. (5), where  $\mathbf{S}$  is a random sensor placement in a rendezvous circle. For each candidate  $p_c$ , 100 random sensor placements are generated (i.e., Monte Carlo trials) and the average  $\Omega$  is used as the information reward relating to  $p_c$ . The second baseline (referred to as PE-MC) is similar to the MI-MC, except that the metric is given by  $\mathbb{H}(\mathbf{S} | \mathbf{H}_c)$ . These two Monte Carlo baselines give the near-optimal swarm centers regarding the MI and PE metrics, respectively. However, due to the high computation overhead of Monte Carlo method, these two baselines are not suitable for mote-class sensor platforms. A random walk approach is employed as the third baseline (referred to as RW). Specifically, the swarm head selects a random position as  $p_c$  subject to the constraints in Eq. (6).

We first visually compare the swarm trajectories scheduled by various approaches. Fig. 9 shows the trajectories of a sensor swarm in the first 6 sampling iterations. Note



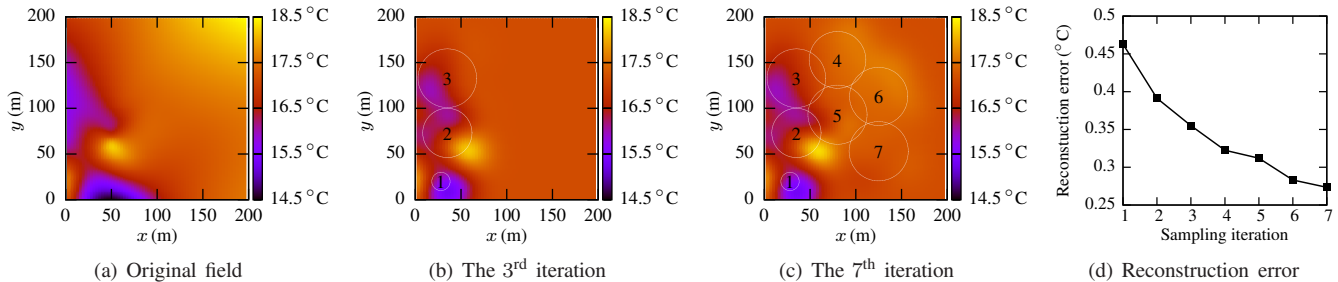


Figure 11. Temperature field reconstruction using a robotic sensor swarm. The numbers in the circles represent the sequence of the rendezvous circles.

that the swarm radius is controlled by the connectivity maintenance algorithm. We can see that, for all approaches, two consecutive rendezvous circles can overlap. This is because the correlation of the Gaussian process exists in both spatial and temporal domains, moving to a farther location does not necessarily increase the overall information reward. Note that, if only spatial correlation is considered, the swarm will move to the farthest unexplored areas.

We then compare the effectiveness of various approaches based on the criterion  $\Omega$  in Eq. (5), which quantifies the drop of uncertainty at the ungauged sites at current time. Fig. 10 plots  $\Omega$  versus the index of sampling iteration. We can see that  $\Omega$  increases over time as more measurements are taken. From the figure, we can see that the MI and MI-MC outperforms the PE by 10% and 19%, respectively, in the 5<sup>th</sup> sampling iteration. However, they have much higher computation overhead than PE. Specifically, MI and MI-MC take about 20 and 8000 times of the execution time of PE, respectively. The RW approach yields the worst accuracy. We also evaluate the two truncation schemes presented in Section V-C. Both *time-trunc* and *cov-trunc* schemes can achieve desirable reconstruction accuracy. In particular, we find that the truncation schemes with  $K = 40$  yield almost the same performance obtained by using all historical measurements. Due to space limitation, the detailed evaluation results are omitted here and can be found in [21].

4) *Accuracy of Field Reconstruction*: In this set of simulations, we reconstruct a field using 10 robotic sensors. The simulations are based on the temperature data [20] collected at 8 locations on the surface of Lake Fulmor, California, which has an area of about 3 acres. Our analysis has verified that the temperature data follow the spatiotemporal Gaussian process [21]. Therefore, we use an existing tool [22] to fit a  $200 \times 200 \text{ m}^2$  ( $\simeq 10$  acres) Gaussian process field based on the traces, as shown in Fig. 11(a). For the ease of illustration, the field does not change with time, although our approach can deal with temporal evolution of the field. The movement of the swarm is scheduled by PE without truncation. Sensor measurements in the simulation are corrupted by zero-mean Gaussian noise with variance of 0.15 [2] [19]. In the reconstruction, the mean function  $\mathcal{M}(p, t)$  is set to be a fixed value of the average temperature in

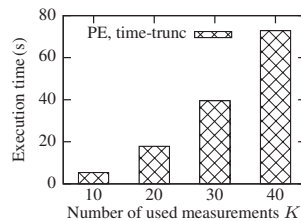


Figure 12. Execution time of PE *time-trunc* scheduling versus the number of used measurements  $K$ .

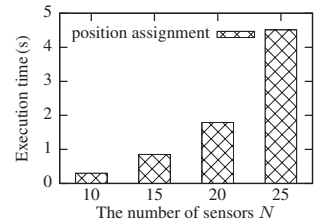


Figure 13. Execution time of sensor position assignment versus the number of sensors  $N$ .

Fig. 11(a). Fig. 11(b) and Fig. 11(c) show the reconstructed field after the 3<sup>rd</sup> and 7<sup>th</sup> sampling iteration, as well as the trajectories of the swarm. Fig. 11(d) plots the average reconstruction error versus the index of sampling iteration. The average reconstruction error is calculated as the average of the absolute difference between the posterior mean given by Eq. (3) and the groundtruth. From Fig. 11, we can see that the accuracy of reconstruction is improved along with the movement of the swarm.

### B. Overhead on Sensor Hardware

We have implemented the PE-based *time-trunc* swarm center selection algorithm and the sensor movement scheduling algorithm in TinyOS 2.1 on TelosB platform. We ported the C implementation of Chelosky decomposition algorithm in GNU Scientific Library [23] to TinyOS to invert matrix in the swarm center selection algorithm. We also implement the Munkres algorithm in TinyOS to schedule each sensor's movement. Fig. 12 and Fig. 13 plot the execution times of the two algorithms in one sampling iteration, respectively. We can see that the PE-based *time-trunc* algorithm takes about one minute when 40 historical measurements are used. The Munkres algorithm for position assignment only takes 4.5 seconds when 25 robotic sensors are deployed. As our current implementation employs extensive floating-point computation, the above processing delays can be further reduced by using fixed-point arithmetic. Nevertheless, a delay of about one minute is acceptable since the duration of each sampling iteration can be much longer than that. Note that the MI metric and the *cov-trunc* scheme result in very long processing delays on TelosB platform because of

large search space and repeated matrix inversion operations. Therefore, they are only suitable for more powerful sensor network platforms such as Imote2 [24].

## VII. CONCLUSION

In this paper, we propose a novel cyber-physical approach to spatiotemporal aquatic field reconstruction using inexpensive, low-power, mobile sensor swarms. Our approach features a rendezvous-based mobility control scheme where a sensor swarm collaborates to sense the environment in a series of carefully chosen rendezvous regions. We design a novel feedback control algorithm that maintains the desirable level of wireless connectivity of a sensor swarm in the presence of significant physical dynamics. We present new information-theoretic analysis to guide the selection of rendezvous regions such that the field reconstruction accuracy is maximized. Extensive trace-driven simulations validate the effectiveness of our approach. The implementation of the algorithms on TelosB mote shows that our approach incurs low overhead on resource-constrained sensor platforms.

## ACKNOWLEDGMENT

This research was supported in part by the U.S. National Science Foundation under grants ECCS-1029683, CNS-0954039 (CAREER), IIS-0916720, and ECCS-1050236.

## REFERENCES

- [1] J. Dolan, G. Podnar, S. Stancliff, E. Lin, J. Hosler, T. Ames, J. Moisan, T. Moisan, J. Higinbotham, and A. Elfes, "Harmful algal bloom characterization via the telesupervised adaptive ocean sensor fleet," in *NASA Science Technology Conference*, 2007.
- [2] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Trans. Robot.*, vol. 27, no. 99, 2011.
- [3] C. Detweiler, M. Doniec, M. Jiang, M. Schwager, R. Chen, and D. Rus, "Adaptive decentralized control of underwater sensor networks for modeling underwater phenomena," in *SenSys*, 2010.
- [4] S. A. Ruberg, R. W. Muzzi, S. B. Brandt, J. C. Lane, T. C. Miller, J. J. Gray, S. A. Constant, and E. J. Downing, "A wireless internet-based observatory: The Real-time Coastal Observation Network (ReCON)," in *OCEANS*, 2007.
- [5] Science Daily, "Researchers experiment with solar underwater robots," 2004, available at <http://sciencedaily.com/releases/2004/12/041212081548.htm>.
- [6] D. L. Rudnick, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry, "Underwater gliders for ocean research," *J. Marine Technology Society*, vol. 38, 2004.
- [7] X. Tan, "Autonomous robotic fish as mobile sensor platforms: Challenges and potential solutions," *J. Marine Technology Society*, vol. 45, no. 4, 2011.
- [8] <http://blooms.uwcfi.org/mendota>.
- [9] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *IPSN*, 2006.
- [10] B. Zhang and G. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *ICRA*, 2007.
- [11] A. Singh, R. Nowak, and P. Ramanathan, "Active learning for adaptive mobile sensing networks," in *IPSN*, 2006.
- [12] A. Singh, A. Krause, C. Guestrin, and W. Kaiser, "Efficient informative sensing using multiple robots," *J. Artificial Intelligence Research*, vol. 34, no. 1, 2009.
- [13] M. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *CDC*, 2006.
- [14] R. Maheshwari, S. Jain, and S. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *SenSys*, 2008.
- [15] T. Adbelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu, *Performance Modeling and Engineering*. Springer, 2008, ch. Introduction to Control Theory and its Application to Computing Systems.
- [16] T. Rappaport, *Wireless communications: principles and practice*. Prentice Hall, 1996.
- [17] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *MobiSys*, 2008.
- [18] K. Ogata, *Discrete-time control systems*. Prentice Hall, 1995.
- [19] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: theory, efficient algorithms and empirical studies," *J. Mach. Learn.*, vol. 9, 2008.
- [20] <http://robotics.usc.edu/~namos/>.
- [21] Y. Wang, R. Tan, G. Xing, X. Tan, J. Wang, and R. Zhou, "Spatiotemporal aquatic field reconstruction using robotic sensor swarm," CSE Department, Michigan State University, Tech. Rep., 2012.
- [22] <http://mathworks.com/matlabcentral/fileexchange/31485>.
- [23] <http://gnu.org/software/gsl>.
- [24] Crossbow Technology, "Imote2 datasheets."