

# Spatiotemporal Aquatic Field Reconstruction Using Cyber-Physical Robotic Sensor Systems

YU WANG, RUI TAN, GUOLIANG XING, XIAOBO TAN, JIANXUN WANG,  
and RUOGU ZHOU, Michigan State University

Monitoring important aquatic processes like harmful algal blooms is of increasing interest to public health, ecosystem sustainability, marine biology, and aquaculture industry. This article 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.

Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems; C.4 [Performance of Systems]: Measurement techniques, Modeling techniques

General Terms: Measurement, Performance

Additional Key Words and Phrases: Robotic sensor swarm, field reconstruction, connectivity control, movement scheduling

## ACM Reference Format:

Yu Wang, Rui Tan, Guoliang Xing, Xiaobo Tan, Jianxun Wang, and Ruogu Zhou. 2014. Spatiotemporal aquatic field reconstruction using cyber-physical robotic sensor systems. *ACM Trans. Sensor Netw.* 10, 4, Article 57 (April 2014), 27 pages.

DOI: <http://dx.doi.org/10.1145/2505767>

## 1. INTRODUCTION

Monitoring aquatic environment is of great interest to public health, ecosystem sustainability, marine biology, and aquaculture industry. In this article we explore an important problem in aquatic monitoring, namely, reconstruction of spatiotemporal

---

A preliminary version of this work was published in *Proceedings of the 33<sup>rd</sup> IEEE Real-Time Systems Symposium (RTSS12)* [Wang et al. 2012]. R. Tan is now with Advanced Digital Sciences Center, Illinois at Singapore.

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

Authors' addresses: Y. Wang (corresponding author), R. Tan, and G. Xing, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824; email: [wangyu3@msu.edu](mailto:wangyu3@msu.edu); X. Tan and J. Wang, Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824; R. Zhou, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2014 ACM 1550-4859/2014/04-ART57 \$15.00

DOI: <http://dx.doi.org/10.1145/2505767>

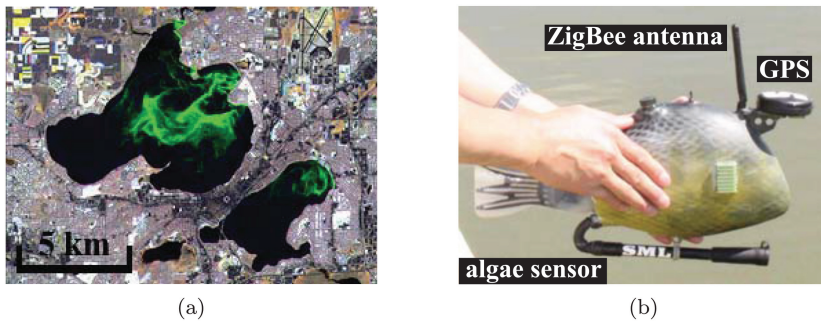


Fig. 1. (a) HABs in Lake Mendota (top left) and Lake Monona (right bottom) in Wisconsin, 1999 [HABs and Lake Mendota 2012] (photo credit: Space Science and Engineering Center at University of Wisconsin-Madison and WisconsinView); (b) a prototype of autonomous robotic fish developed by the Smart Microsystems Laboratory at Michigan State University [Tan 2011].

aquatic process. Many physical and biological phenomena in an aquatic environment, including Harmful Algal Blooms (HABs) [Dolan et al. 2007], lake surface temperature [Xu et al. 2011], and plume concentration of chemical substance [Detweiler et al. 2010], can be modeled as spatiotemporal aquatic fields that usually follow certain distributions such as the spatiotemporal Gaussian process. For instance, Figure 1(a) shows the HABs on 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 environments. This approach is labor intensive and has difficulty in capturing large-scale spatially distributed phenomena of interest. An alternative approach is in-situ sensing with fixed or buoyed/moored sensors [Ruberg et al. 2007]. 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) [Science Daily 2004] and sea gliders [Rudnick et al. 2004] 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 article we propose to use inexpensive, low-power robotic sensor platforms to sample and reconstruct spatiotemporal aquatic processes of interest. Figure 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 [Tan 2011], and can be interfaced with various aquatic sensors. Robotic fish can form an autonomous network and sense an 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 to address these challenges.

- (1) We propose a new approach to the sampling and reconstruction of a 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 robustly maintain network connectivity 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 article is organized as follows. Section 2 reviews related work. Section 3 introduces the background and provides an overview of our approach. Section 4 presents the control-theoretic connectivity maintenance algorithm. Section 5 presents the information-theoretic swarm movement scheduling algorithms. Section 6 discusses several issues and the possible extensions to this work. Section 7 presents the results of extensive trace-driven simulations and implementation on a sensor platform. Section 8 concludes this work.

## 2. RELATED WORK

Sampling and reconstruction of physical fields using networked sensor systems has recently received increasing interest. Early work focuses on stationary sensor deployment. In Krause et al. [2006], positions of sensors are selected before 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. A fast sensor placement approach for fusion-based field surveillance is proposed in Chang et al. [2011] to minimize the number of sensors while maintaining the signal-to-noise ratio. Recently, mobility has been exploited to enhance the adaptability and sensing capability of sensor systems. In Zhang and Sukhatme [2007], a robotic boat supplements a static sensor network to reduce the error of field reconstruction, where the boat's movement is guided by the measurements of the sensor network. Another study [Singh et al. 2006] develops active learning schemes for mobile sensor networks, which plan the movements of mobile sensors based on the feedback of previous measurements. In our previous

work [Wang et al. 2012], we develop movement scheduling algorithms for a school of robotic fish to profile aquatic diffusion processes. Several recent studies focus on leveraging sensors' mobility to reconstruct physical fields that follow the Gaussian process. In Xu et al. [2011], the movement of mobile sensors is directed to reduce the uncertainties in estimating the field variables at a set of prespecified locations. The algorithms developed for placing stationary sensors in Krause et al. [2006] are extended to schedule the movement of a mobile sensor network in reconstructing a Gaussian process [Singh et al. 2009]. However, the aforementioned studies do not account for the constraints of low-power robotic sensor systems, such as the limited motion, computation, and communication capabilities. Moreover, they generally focus on the open-loop solutions that often fail to adapt to the highly complex and dynamic aquatic environment.

The Gaussian process field reconstruction using mobile sensor networks has also been extensively studied in Cortés [2009], Low et al. [2008, 2009, 2011, 2012], and Chen et al. [2012]. In Cortés [2009], the movements of mobile sensors are controlled to follow the gradient ascent directions of the Gaussian process field to increase the information reward. In Low et al. [2009], an adaptive path planning approach is presented for mobile sensors in exploring and mapping the hotspot fields. However, this centralized approach can incur heavy computation overhead if the number of observations or sensors is large. To improve the computation efficiency, a decentralized approach is designed in Chen et al. [2012], with consideration of the limited communication capability of mobile sensors. For more studies on Gaussian process field reconstruction using mobile sensors, we refer the interested reader to Low et al. [2008, 2011, 2012], and Chen et al. [2012] and the references therein. Different from these existing studies that typically focus on improving certain aspects of the reconstruction problem, in this work we aim to develop a practical and integrated approach based on a swarm scheme, which jointly addresses limited mobility and processing capability of the robotic sensor, as well as the dynamic on-water wireless link quality. In our approach, the computation of swarm movement scheduling and field reconstruction is executed at the swarm head. The computation efficiency of our approach can be improved by integrating the decentralized/distributed field reconstruction and sensor movement scheduling algorithms in Cortés [2009], Low et al. [2012], and Chen et al. [2012].

Most previous works on maintaining sensor network connectivity adopt the graph theory [Xu et al. 2011] and the potential field theory [De Gennaro and Jadbabaie 2006], and assume fixed communication range and reliable communication quality. However, several studies have revealed significant stochasticity and irregularity in link quality of low-power wireless sensors [Zuniga and Krishnamachari 2004; Qiu et al. 2007; Maheshwari et al. 2008; Chen and Terzis 2011]. Feedback control has been widely adopted to improve the adaptability of computing systems [He et al. 2003; Lin et al. 2006; Adbelzaher et al. 2008; Liu et al. 2010]. Different from these existing solutions, our control-theoretic connectivity maintenance algorithm specifically deals with the dynamics caused by movement of a robotic sensor swarm and disturbances from the aquatic environment. Mobility has been used to improve link quality and preserve network connectivity for robotic sensor systems. In Twigg et al. [2012], each robotic sensor moves in the gradient ascent direction of its received signal strength (RSS). However, the movement scheduling algorithm developed in Twigg et al. [2012] considers only a single link. Moreover, to obtain an estimate of RSS gradient, the robotic sensor has to explore the local area, which increases energy consumption in movements. In this article we propose a feedback-control-based approach that aims to adaptively maintain the network connectivity of a robotic sensor swarm in the presence of various environment and system dynamics. Our control-theoretic algorithm does not require the energy-consuming exploration in local area.

Recently, several swarm-based CPSs have been proposed for various sensing applications. Representative examples include RoboBee [Dantu et al. 2011] and SensorFly [Purohit et al. 2011]. These studies mainly focus on hardware design and system issues. In contrast, this article addresses the field reconstruction problem using a robotic sensor swarm. Based on key observations from real data traces of robotic sensors' wireless communication and field measurements, we formulate the swarm connectivity control and movement scheduling problems, and solve them using control- and information-theoretic algorithms.

### 3. OVERVIEW OF APPROACH

#### 3.1. Background and Challenges

Our objective is to reconstruct an aquatic scalar field that follows the spatiotemporal Gaussian process using a group 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 [Tan 2011], as shown in Figure 1(b). These robotic sensor platforms are typically equipped with computation, communication, movement control, GPS components, as well as various sensors [Tan 2011]. However, due to the resource constraints, they have limited capabilities of computation, communication, and movement. For instance, the TelosB mote integrated with the robotic fish platform shown in Figure 1(b) only has an 8 MHz MCU and a low-power 802.15.4 radio with short communication range. In this article 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 Figure 1(a) [HABs and Lake Mendota 2012]). However, the number of robotic sensors available in practice is often small (e.g., a few dozens). In addition, as the robotic sensors in the aquatic environment often have short communication ranges, the area that a networked 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.

#### 3.2. Approach Overview

A simple approach to reconstructing the field using robotic sensors is to send sensors to regions that evenly divide the whole aquatic field and each sensor only samples its own region. Because the aquatic process typically covers a large area as discussed in Section 3.1, under this simple approach, the sensors would not be able to communicate with each other. Therefore, this noncollaborative approach has the following two drawbacks. First, each sensor can only reconstruct the field based on its own measurements, and the field reconstruction based on all sensor measurements cannot be performed until sensors complete their sampling and gather at some location. Second, the accuracy of the whole field reconstruction would be significantly undermined if some sensors experience failures.

To address the challenges discussed in Section 3.1, we adopt a novel *rendezvous-based swarm* scheme as illustrated in Figure 2. We assume that all sensors know their positions and are time synchronized, for example, through GPS or in-network

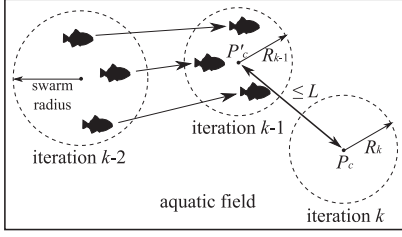


Fig. 2. Rendezvous-based swarm scheme. Dashed circles represent the rendezvous circles.

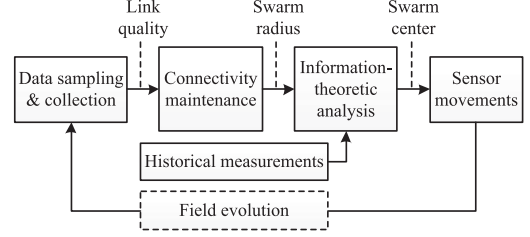


Fig. 3. The iterative sampling process of a robotic sensor swarm.

localization/synchronization services. The robotic sensor system iteratively samples the aquatic field. As shown in Figure 3, in each sampling iteration, robotic 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, as well as schedules the movements of sensors in the next sampling 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. In our approach, the movement scheduling at the swarm head is executed as follows.

- (1) The swarm head first assesses the quality of 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, such that the network connectivity in the next sampling iteration can achieve a desirable level.
- (2) Given the projected swarm radius, the swarm head conducts information-theoretic analysis to select the location of the next rendezvous circle, in order to maximize the improvement of the field reconstruction accuracy.
- (3) The swarm head generates random target positions within the next rendezvous circle and assigns the positions to each sensor to minimize the total movement distance. The target positions are finally sent to the sensors. Under this random target position approach, small motion control errors can be tolerated as long as the final positions of sensors fall within the rendezvous circle. Moreover, as proved in Section 5.5, under this approach, there is no crossing between sensors' moving paths and hence the robotic sensors would not collide.

After receiving the target position in the next sampling iteration, each sensor straightly moves toward its destination to minimize the energy consumption of locomotion. To initiate the preceding process, the swarm is initially dropped at a venue within the region affected by the physical/biological process of interest. Note that, to balance the energy consumption of sensors, the swarm head role can rotate among all sensors. The communication overhead of our approach is low because sensors coordinate with each other only when they gather in a rendezvous circle. In summary, our swarm scheme allows the robotic sensors to efficiently collaborate in sensing a large dynamic aquatic field and avoid heavy coordination overhead. Therefore, it is practical and energy efficient for low-power aquatic robotic platforms [Tan 2011; Zhang and Sukhatme 2007].

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

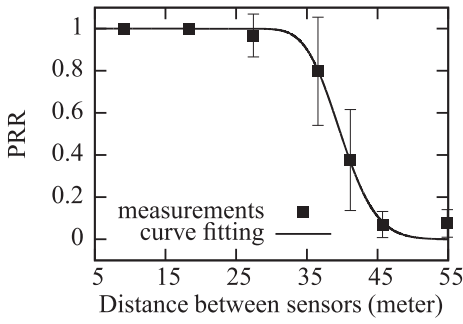


Fig. 4. The PRR measurements versus the distance between two sensors. The error bar represents standard deviation.

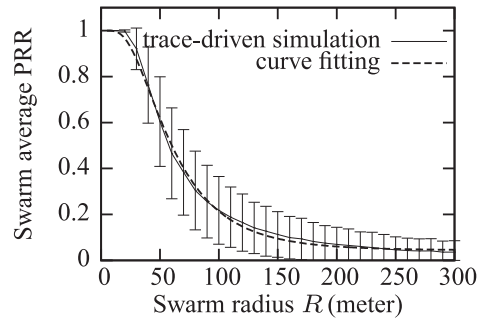


Fig. 5. The swarm average PRR versus the swarm radius. The error bar represents the standard deviation.

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 procedure 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 the energy budget, such that the improvement of the field reconstruction accuracy can be maximized with the newly obtained sensor measurements. In this article 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.

#### 4. SWARM CONNECTIVITY MAINTENANCE

The wireless connectivity between a robotic sensor and the swarm head is affected by various 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. In this section, we first study the on-water wireless link dynamics based on real data traces collected on a lake. We then analyze the swarm connectivity. Finally, we formulate the connectivity maintenance as a feedback control problem that aims to maintain the swarm connectivity at a desired level by adjusting the swarm radius based on the quality of all links measured at runtime.

##### 4.1. On-Water Wireless Link Dynamics

We first motivate our approach using PRR traces of an on-water 802.15.4 wireless link. Figure 4 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 IRIS 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. From Figure 4, we have the following two important observations. First, the on-water wireless communication has a limited reliable communication range, which is about 35 m for a typical 802.15.4 radio. According to our experience, the communication range of IRIS mote on water surface decreases by about 50% compared to that on land. Second, the PRR shows significant variance, especially in the transition range from 25 m to 40 m. It is mostly caused by the radio and environment dynamics [Zuniga and Krishnamachari 2004]. The wireless link in a wavy water environment is more dynamic than that in a calm water environment, due to the multipath effect and fading. Such highly dynamic communication quality can lead to increased communication cost in the field sampling, and even loss of sensors due to a disconnected network. Therefore, it is critical to maintain satisfactory connectivity under radio and environment dynamics.

## 4.2. Modeling Swarm Connectivity

As discussed in Section 3.2, the sensor swarm forms a network with single-hop star topology in a rendezvous circle. Compared with multihop topology, the single-hop topology of the sensor swarm incurs significantly lower overhead in communication and network formation/maintenance. Suppose the reliable on-water communication range of a typical 802.15.4 radio is 35 m. Under the single-hop star topology centered at the swarm head, 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. In this section, we first derive the expression for the average PRR given swarm radius, which allows us to adaptively control the swarm connectivity by adjusting the swarm radius. We then verify the closed-form expression using real data traces.

**4.2.1. Model Derivation.** 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  $P_r(d) = P_t - PL(d_0) - 10\alpha \log_{10}(d/d_0)$  [Rappaport 1996], 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  $\xi^2$  [Rappaport 1996]. 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  $\eta$  [Judd et al. 2008]. Hence, the PRR of a single link can be derived as

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

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.

Based on the single-link PRR model given in Eq. (1), 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 consists of 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



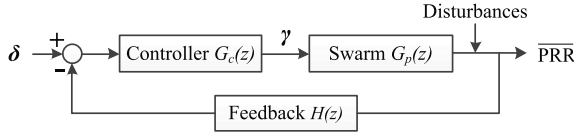


Fig. 6. The closed loop for connectivity control.

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.

**4.2.2. Model Validation.** We now use the collected PRR traces of on-water wireless communication (see Section 4.1) to verify the preceding models. We start from the link PRR model given in Eq. (1). 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 plotted in Figure 4. 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. Although Eq. (1) only captures the expected PRR, the control-theoretic connectivity maintenance algorithm presented in Section 4.3 accounts for the variance of PRR measurements.

We then conduct Monte Carlo simulations to verify the accuracy of the swarm average PRR model given in Eq. (2), 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. Figure 5 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 the wireless link. We then fit the curve defined by Eq. (2) with the simulation results, as shown in Figure 5. 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 performance evaluation in Section 7.

### 4.3. 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 Figure 5, 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  $\delta$ , such as 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 Figure 6. We denote  $G_c(z)$ ,  $G_p(z)$ , and  $H(z)$  as the transfer functions of the connectivity maintenance algorithm, the sensor swarm system, and the feedback, which are expressed in  $z$ -transform

representation. The  $z$ -transform provides a compact representation for time-varying functions, where  $z$  represents a time shift operation. We refer the interested reader to Ogata [1995] for the details of  $z$ -transform and Abdelzaher et al. [2008] for a few representative applications of discrete-time control theory to networking and computing systems. As shown in Figure 6, 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  $R$  (refer to 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, we have the swarm average PRR expressed as  $\overline{\text{PRR}}(\gamma) \simeq (1 - c) + c \cdot \gamma$ . As this time-domain expression does not contain time shift, its  $z$ -transform is simply  $G_p(z) = c$  [Ogata 1995]. 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 [Ogata 1995]. Hence, we let  $G_c(z) = \frac{\alpha}{1 - \beta z^{-1}}$ , 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 [Ogata 1995], the stability and convergence condition can be obtained as  $\beta = 1$  and  $0 < \alpha < 2/c$ . The detailed analysis can be found in Appendix A.1.

In this article we model three uncertainties that substantially affect the  $\overline{\text{PRR}}(R)$  as the disturbances in the control loop shown in Figure 6. First, as shown in Figure 4, 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 Figure 5 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 [Ogata 1995], 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 article,  $b$  is set to be 0.9.

Implementing  $G_c(z)$  in the time domain gives the connectivity maintenance algorithm. According to Figure 6, 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  $R_k = 10^{(\text{erf}^{-1}(\gamma_k) - c_2)/c_1}$ .

## 5. 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.

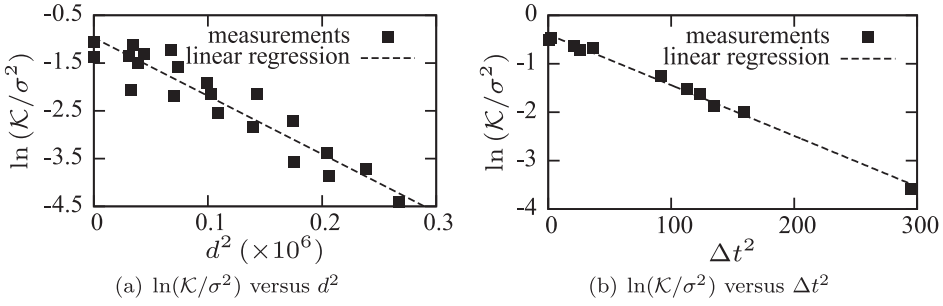


Fig. 7.  $\ln(K/\sigma^2)$  versus  $d^2$  and  $\Delta t^2$ .

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.

### 5.1. Physical Field

**5.1.1. Spatiotemporal Gaussian Process Model.** We assume that the monitored physical phenomenon follows the spatiotemporal Gaussian process [Rasmussen 2006]. 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 article we adopt the following covariance function that has been widely adopted [Dolan et al. 2007; Xu et al. 2011; Krause et al. 2006]

$$\mathcal{K}(d, \Delta t) = \sigma^2 \cdot \exp\left(-\frac{d^2}{2\varsigma_s^2}\right) \cdot \exp\left(-\frac{\Delta t^2}{2\varsigma_t^2}\right), \quad (3)$$

where  $d = \|p - p'\|_{\ell_2}$ ,  $\Delta t = |t - t'|$ ,  $\sigma^2$  is the prior variance of any field variable and  $\varsigma_s$  and  $\varsigma_t$  are the spatial and temporal *kernel bandwidths*, respectively. 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, that is,  $\mathbf{Z} \sim \mathcal{N}(\mathbf{m}, \Sigma)$ , 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 [Xu et al. 2011]. 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  $\sigma_w^2$ .

**5.1.2. Model Verification.** We now verify the Gaussian process model using real temperature traces collected on Lake Fulmor, California [NAMOS Project 2006]. The temperature readings on the lake surface were collected by 8 robotic boats over several hours. Applying the logarithm to the covariance function  $\mathcal{K}(d, \Delta t)$  yields  $-2 \cdot \ln \mathcal{K}(d, \Delta t)/\sigma^2 = d^2/\varsigma_s^2 + \Delta t^2/\varsigma_t^2$ . Therefore, the quantities  $\ln(K/\sigma^2)$ ,  $d^2$ , and  $\Delta t^2$  are expected to exhibit linear relationships. Figure 7 plots  $\ln(K/\sigma^2)$  versus  $d^2$  and  $\Delta t^2$ , respectively. We can observe from the figure that the quantities exhibit linear relationships with small variations caused by the random noise  $W$ . The hyperparameters are estimated as  $\varsigma_s = 6.42$  and  $\varsigma_t = 7.15$ . Therefore, the adopted  $\mathcal{K}(d, \Delta t)$  well characterizes the spatiotemporal covariance of the water surface temperatures.

## 5.2. Field Reconstruction Using a Robotic Sensor Swarm

In this section, we present how to reconstruct the field using measurements collected by the sensor swarm. To facilitate the expression, we define  $\mathbf{H}$  as a row vector composed of all measurements, namely,  $\mathbf{H} = [R(p_1, t_1), \dots, R(p_N, t_N)]$ ,  $\mathbf{m}$  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 a multiple of  $T$ . The  $\mathbf{H}_c$  is a  $3 \times N$  matrix, where each column represents 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 denoted by  $\mathbb{E}[Z|\mathbf{H}]$  and  $\text{Var}[Z|\mathbf{H}]$ . The estimates are given by Rasmussen [2006] and Ramachandran and Tsokos [2009] as

$$\mathbb{E}[Z|\mathbf{H}] = \mathcal{M}(p, t) + \tilde{\Sigma}[(p, t), \mathbf{H}_c] \cdot \tilde{\Sigma}^{-1}[\mathbf{H}_c] \cdot (\mathbf{H} - \mathbf{m})^T, \quad (4)$$

$$\text{Var}[Z|\mathbf{H}] = \sigma^2 - \tilde{\Sigma}[(p, t), \mathbf{H}_c] \cdot \tilde{\Sigma}^{-1}[\mathbf{H}_c] \cdot \tilde{\Sigma}^T[(p, t), \mathbf{H}_c], \quad (5)$$

where  $\tilde{\Sigma}$  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 Eq. (4) and Eq. (5). First, because of the spatiotemporal correlation, the posterior variance (i.e., the uncertainty) is reduced given the measurements  $\mathbf{H}$ . Second, from Eq. (5), 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, the dimension of  $\tilde{\Sigma}$  increases along the accumulation of sensor measurement. As a result, the high-dimensional  $\Sigma$  poses substantial computation overhead to calculate its inversion in Eq. (4) and Eq. (5) on resource-constrained robotic sensors.

From the previous three observations, an important design of our approach is to separate the following two tasks.

*Sensor movement scheduling.* This task is executed on the swarm head in each sampling iteration, which aims to reduce the variance given in Eq. (5). Section 5.3 to Section 5.5 will present the details of our sensor movement scheduling algorithms. In particular, as the sensor movement scheduling involves calculating the inversion of  $\tilde{\Sigma}$  in Eq. (5) on the swarm head, in Section 5.4, we propose two measurement truncation schemes that can significantly reduce the computation overhead.

*Field reconstruction.* This task computes Eq. (4) and Eq. (5) based on collected measurements. It can be executed on either on the swarm head, if it has sufficient computation capability, or a remote data processing center after measurements are fetched back.

## 5.3. 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., Eq. (4) and Eq. (5)).

**5.3.1. Problem Formulation.** 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 the time of reconstruction, 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

<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 swarm center in the  $(k - 1)^{\text{th}}$  and  $k^{\text{th}}$  sampling iteration, and  $R_k$  is the scheduled swarm radius for the  $k^{\text{th}}$  iteration by the connectivity maintenance algorithm. The optimal solution of  $\mathbf{S}$  maximizes the following information-theoretic metric

$$\Omega(\mathbf{S}) = \mathbb{H}[\mathbf{V} \setminus \mathbf{S} | \mathbf{H}_c] - \mathbb{H}[\mathbf{V} \setminus \mathbf{S} | \mathbf{H}_c \cup \mathbf{S}], \quad (6)$$

subject to

$$\begin{aligned} \|p'_c - p_c\|_{\ell_2} &\leq L; \\ \|p_i - p_c\|_{\ell_2} &\leq R_k, \forall i \in [1, N]; \end{aligned} \quad (7)$$

where the  $\mathbb{H}[\cdot]$  denotes entropy and quantifies the uncertainty. In Eq. (6), the term  $\mathbf{V} \setminus \mathbf{S}$  represents the set of ungauged sites in the current iteration, and the term  $\mathbf{H}_c \cup \mathbf{S}$  represents the set of visited time-space coordinates after the current iteration. Therefore,  $\mathbb{H}[\mathbf{V} \setminus \mathbf{S} | \mathbf{H}_c]$  represents the uncertainty at the ungauged sites (i.e.,  $\mathbf{V} \setminus \mathbf{S}$ ) given the historically visited positions, and  $\mathbb{H}[\mathbf{V} \setminus \mathbf{S} | \mathbf{H}_c \cup \mathbf{S}]$  represents the uncertainty at the ungauged sites after additionally sampling the field at  $\mathbf{S}$ . As a result, the preceding problem aims to maximize the drop of entropy at the ungauged sites after the current iteration by sampling the field variables at  $\mathbf{S}$  given the historical measurements at  $\mathbf{H}_c$ . The previous problem formulation adopts the drop of entropy as the performance metric, which is defined by Eq. (6). The constraint in the first part of Eq. (7) 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. (7) ensures the scheduled swarm radius. These constraints are also illustrated in Figure 2.

Note that the posterior entropy for the ungauged sites [Wang et al. 2004] is another widely adopted performance metric in field reconstruction studies. We now identify the relationship between posterior entropy and our metric defined in Eq. (6). Suppose the current iteration is the  $k^{\text{th}}$  iteration of the sampling process. By cumulating Eq. (6) of each iteration, we can approximate the accumulative entropy reduction (denoted as  $\sum_{i=1}^k \Omega(\mathbf{S}_i)$ ) as  $\sum_{i=1}^k \Omega(\mathbf{S}_i) \approx \mathbb{H}[\mathbf{V} \setminus \mathbf{S}_1] - \mathbb{H}[\mathbf{V} \setminus \mathbf{S}_k | \mathbf{H}_c \cup \mathbf{S}_k]$ , where  $\mathbf{S}_i$  is the set of sampling positions in the  $i^{\text{th}}$  iteration and  $\mathbf{H}_c = \mathbf{S}_1 \cup \mathbf{S}_2 \dots \cup \mathbf{S}_{k-1}$  represents all the gauged sites until the  $k^{\text{th}}$  iteration. In particular, the term  $\mathbb{H}[\mathbf{V} \setminus \mathbf{S}_k | \mathbf{H}_c \cup \mathbf{S}_k]$  denotes the posterior entropy after the  $k^{\text{th}}$  iteration. Note that  $\mathbb{H}[\mathbf{V} \setminus \mathbf{S}_1]$  is a constant. Therefore, there is a simple linear relationship between the posterior entropy and the accumulated entropy reduction. From this relationship, the minimum posterior entropy is achieved when the entropy reduction in each iteration is maximized.

**5.3.2. Swarm Center Selection Algorithm.** A similar problem without the condition  $\mathbf{H}_c$  and the constraints in Eq. (7) has been proven NP-hard [Krause et al. 2008]. Hence, the preceding problem has prohibitively high complexity that is not practical for robotic sensor platforms. In this article we propose a heuristic approach that approximates the whole swarm by its center, which is selected from a set of discrete candidate points. By such an approximation, we avoid the complex inter-point dependence given by Eq. (3), hence largely reduce the computation overhead. We will evaluate the performance of this approximation in Section 7.1.8. Under the proposed 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 resulting reconstruction accuracy, they allow the system designer to choose a 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  $\mathbb{I}[X; \mathbf{Y}] = \mathbb{H}[X] - \mathbb{H}[X | \mathbf{Y}]$ , where

$\mathbb{H}[X|\mathbf{Y}] = \frac{1}{2} \log(2\pi e \cdot \text{Var}[X|\mathbf{Y}])$  and

$$\text{Var}[X|\mathbf{Y}] = \text{Var}[X] - \Sigma[X, \mathbf{Y}] \cdot \Sigma^{-1}[\mathbf{Y}] \cdot \Sigma^T[X, \mathbf{Y}]. \quad (8)$$

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

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

The aforesaid information reward metric characterizes the drop of uncertainty about the region other than  $p_c$  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. (7).

The complexity for computing  $\Omega_{\text{MI}}$  for a certain  $p_c$  is  $O(|\mathbf{V}|^3)$ . 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:

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

Different from  $\Omega_{\text{MI}}$ ,  $\Omega_{\text{PE}}$  characterizes the uncertainty drop at the swarm center  $p_c$  in the next iteration given the historical measurements. For each certain  $p_c$ , the complexity of computing  $\Omega_{\text{PE}}$  is  $O(|\mathbf{H}_c|^3)$ , 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.

#### 5.4. 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. (8). This imposes substantial challenges to robotic sensor platforms with limited computation resources. For instance, a TelosB mote equipped with 10KB 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. (8) 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 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

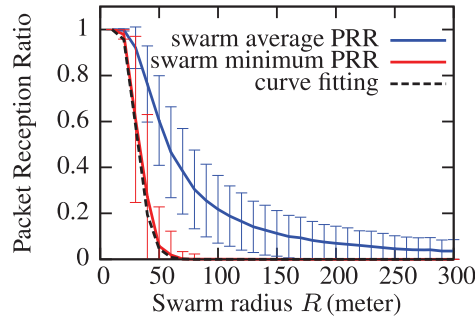


Fig. 8. The swarm average and minimum PRR versus the swarm radius. The error bar represents the standard deviation.

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. However, we note that the performance of these truncation schemes depends on the properties of the underlining aquatic processes, such as the kernel bandwidths (i.e.,  $\zeta_s$  and  $\zeta_t$ ) and the affected region (i.e.,  $\mathbf{V}$ ). In Section 7.1.6, we will evaluate the impact of historical measurements truncation on reconstruction accuracy.

### 5.5. Sensor Movement Scheduling

As discussed in Section 3, once the swarm center and radius are determined, the swarm head randomly selects  $N$  positions (denoted by  $\mathbf{p}'$ ) in the rendezvous circle. We let  $\mathbf{p}$  denote the current positions of robotic sensors. To prolong the lifetime of the robotic sensor swarm, we find the element mapping from  $\mathbf{p}$  to  $\mathbf{p}'$ , such that the sum of sensors' movement distances is minimized. Under this movement scheduling scheme, there is no crossing between sensors' moving paths. The proof can be found in Appendix A.2. Therefore, our movement scheduling scheme is collision free. This element mapping problem can be solved by existing algorithms such as the Munkres assignment algorithm [Burkard et al. 2009] with a complexity of  $O(N^3)$ . Once the mapping is found, the swarm head sends the target position to each robotic sensor, which then moves toward the target position. While moving toward the target position, the robotic sensor can adopt a feedback-based motion control algorithm that adaptively corrects the motion errors based on the localization result, using a potential function approach [Baras et al. 2003]. The motion control of robotic fish is beyond the scope of this article and the details can be found in Baras et al. [2003].

## 6. DISCUSSIONS

### 6.1. Swarm Connectivity Based on Minimum PRR

In Section 4, we employ the swarm average PRR as the metric to characterize the swarm connectivity. We note that it is possible that some individual links may have low PRRs while others have high. To this end, we conduct the following Monte Carlo simulations to analyze the worst case of link connectivity. For a given swarm radius  $R$ , we generate a large number of random sensor placements within the rendezvous circle, and select the minimum PRR among all links in the swarm. The simulation results are shown in Figure 8. In Figure 8, we also include the results shown in Figure 5 such that we can compare the minimum PRR and average PRR. We can see that, when the

expected swarm average PRR is 0.9, the expected minimum PRR is 0.6, which means  $1/0.6 = 1.67$  retransmissions on average. Such an overhead is acceptable. Therefore, by specifying a high setpoint for the average PRR, the minimum PRR can be efficiently lower-bounded.

We further extend our approach to using swarm minimum PRR as the metric to set the radius. In other words, the setpoint of the control-theoretic connectivity maintenance algorithm is the desired minimum single-link PRR in the swarm. Deriving the closed-form formula for the expected minimum PRR is challenging. However, numerical analysis shows that the model given by Eq. (2) can well approximate the swarm minimum PRR. Specifically, Figure 8 shows the result of fitting Eq. (2) with the Monte Carlo simulation results. From the figure, we can see that Eq. (2) accurately characterizes the swarm minimum PRR. Therefore, our control-theoretic connectivity maintenance algorithm can still be applied to maintain swarm minimum PRR at a specified level.

## 6.2. Impact of Connectivity Degradation and Outage

Our approach can tolerate connectivity degradation caused by sensor position errors. In the swarm, sensor positions are randomly selected within the rendezvous circle. As long as the final positions of sensors fall within the target rendezvous circle, the swarm connectivity can be maintained by our control-theoretic algorithm. In case a few sensors are outside the rendezvous circle for a few meters, the overall swarm connectivity will not be substantially jeopardized. According to our experimental results presented in Figure 4, the link PRR in the swarm drops at most 15% when the distance is increased by 2. Note that, based on the measurements in our previous work [Wang et al. 2012], the closed-loop motion control algorithms usually introduce small position errors (in the order of 10 cm) and the GPS localization errors are generally around 2 m in an outdoor environment.

As we typically set a sufficiently high setpoint for the swarm connectivity (i.e.,  $\delta$ ), the swarm is expected to maintain a satisfactory connectivity. However, the control-theoretic algorithm may not be able to cope with sudden drastic drops of link quality due to unexpected wireless communication outages. We now describe two recovery mechanisms to prevent loss of sensors in these wireless communication outages. First, packet acknowledgment should be adopted, and the number of retransmissions in case of packet loss can be set to a relatively large value. This simple approach can largely reduce the possibility of sensor loss caused by suddenly reduced link quality that the control-theoretic connectivity maintenance algorithm cannot deal with. Second, if the first mechanism fails, the sensor swarm can gather at a predefined meeting point. When disconnection with the swarm head is detected, a lost node will move to the nearest meeting point. Note that the meeting points can be carefully chosen before system deployment and stored in each sensor.

## 7. 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 swarm movement scheduling algorithms using extensive simulations based on real data traces of water surface temperature field [NAMOS Project 2006] and on-water ZigBee wireless communication. Second, we implement one of the proposed swarm movement scheduling algorithms on a TelosB sensor platform and evaluate its overhead. The results provide insights into the feasibility of adopting advanced information-theoretic movement scheduling algorithms on mote-class robotic sensor platforms.



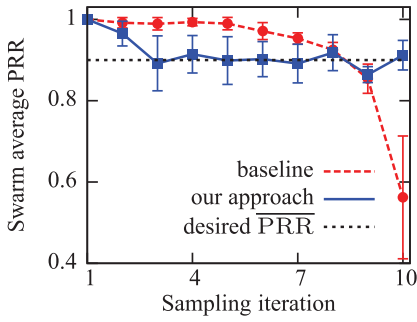


Fig. 9. Swarm average PRR versus sampling iteration. The error bar represents the standard deviation.

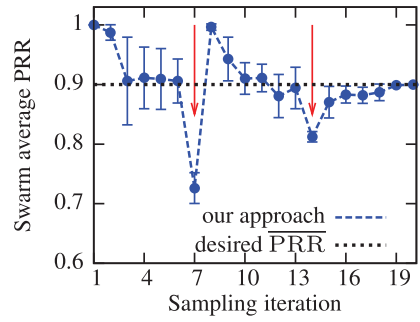


Fig. 10. Impulsive and step responses (at the 7<sup>th</sup> and 14<sup>th</sup> iterations) of our algorithm.

## 7.1. Trace-Driven Simulations

**7.1.1. 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  $[\sigma^2, \zeta_s, \zeta_t] = [9, 6, 8]$ , unless otherwise specified. Note that these settings are consistent with Zhang and Sukhatme [2007] and Singh et al. [2009] and obtained from real on-water temperature traces [NAMOS Project 2006]. 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 (refer to Section 4.2). 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.

**7.1.2. Swarm Connectivity Maintenance and Communication Overhead.** We first compare our connectivity maintenance algorithm with a heuristic baseline algorithm. The heuristic algorithm adopts the 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). Recall that our approach assigns a fixed value to  $c$  and tunes the swarm radius directly. Figure 9 plots the  $\overline{\text{PRR}}(R)$  in the first 10 sampling iterations. The range of swarm radius after 10 iterations is [24, 36]. The error bars, calculated from 20 runs, are caused by the various disturbances discussed in Section 4.3. We can see that the swarm average PRR controlled by our algorithm quickly converges to the desired connectivity level. In contrast, the heuristic algorithm diverges from the reference. This is because the Kalman filter does not tune the swarm radius directly, and incorrectly updates the coefficient  $c$  in the control cycle. To evaluate the response of our algorithm to the sudden changes of the wireless link quality, we artificially reduce the PRR measurements by 20% only in the 7<sup>th</sup> iteration (i.e., the left arrow in Figure 10) and continuously reduce the PRR measurements by 10% after the 14<sup>th</sup> iteration (i.e., the right arrow in Figure 10). For both types of changes, our algorithm can converge within a few iterations.

In each sampling iteration, the communication overhead is mainly caused by the packet loss. Hence, we employ the total number of transmissions in collecting all sensor measurements as the evaluation metric. When a node transmits a packet to the swarm head, the packet is delivered with a success probability equal to the PRR. The node retransmits the packet up to 20 times before it is dropped. The packet to the swarm head includes sensor ID, spatiotemporal coordinates, and measurement. The packet to the sensor contains the target position in the next rendezvous circle. Consider a

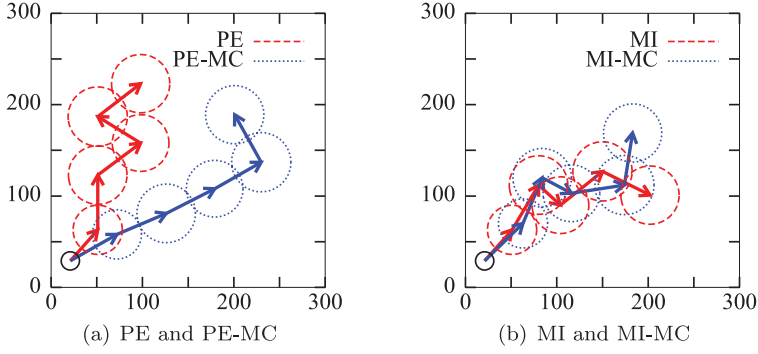


Fig. 11. Trajectories of a robotic sensor swarm with 10 sensors in the first 6 sampling iterations in the reconstruction of a  $300 \times 300 \text{ m}^2$  field.

typical sampling iteration, for example, the 4<sup>th</sup> iteration in Figure 10 where a swarm radius around 32 m yields a swarm average PRR about 0.9. Our simulation results show that for a swarm consisting of 10 nodes, the number of transmitted packets (for two-way communications) has a mean of 38 and a standard deviation of 8. Even if all these packets are transmitted sequentially, the delay will be within a second, because transmitting a TinyOS packet only takes about 10 milliseconds on typical mote-class sensor platforms. Therefore, our approach has low communication overhead.

**7.1.3. Effectiveness of Swarm Center Selection.** We now compare the two swarm center selection approaches presented in Section 5.3.2 (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. (6), where  $\mathbf{S}$  is a set of random sensor placements within the 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 the 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. (7).

We first show the swarm trajectories scheduled by various approaches. Figure 11 plots the trajectories of a sensor swarm in the first six sampling iterations. Note 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. From Figure 11(a), it can be seen that PE and PE-MC output different trajectories. As the next swarm location is affected by historical sensor positions which were randomly generated, the trajectories can be different under different approaches, and even different under the same approach in different simulation runs. However, they basically follow the similar trend of spreading out in the field.

We then compare the effectiveness of various approaches based on the criterion  $\Omega$  given in Eq. (6), which quantifies the drop of uncertainty at the ungauged sites at the current time. Note that in each sampling iteration, the position of the rendezvous circle is determined by the specified information reward metric, and the radius is chosen

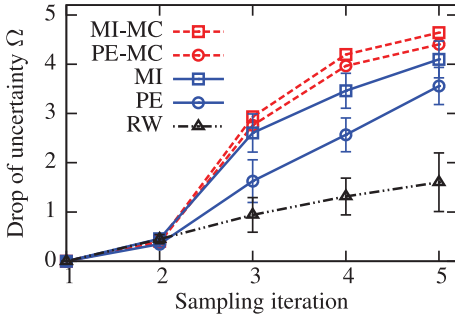


Fig. 12. Information reward versus sampling iteration.

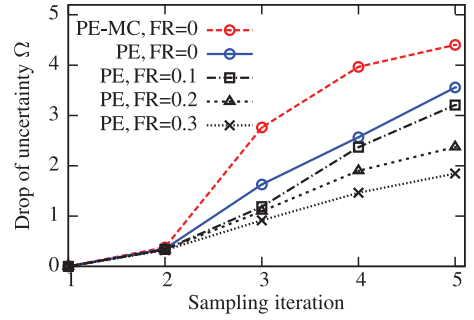


Fig. 13. Impact of sensing failure on information reward (FR stands for failure rate).

to maintain the swarm connectivity. Figure 12 plots  $\Omega$  versus the index of sampling iteration. The error bar represents the standard deviation over multiple simulation runs. We can see that  $\Omega$  increases over time as more measurements are taken. From the figure, we find that MI and MI-MC outperforms 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. Moreover, the gap between our approach and the corresponding Monte-Carlo-based baseline (e.g., PE and PE-MC) gives the performance loss caused by approximating the rendezvous circle with the swarm head. Due to the large number of Monte Carlo trials, the baseline approaches achieve better performance with substantially heavier computation overhead that is infeasible on mote-class platforms.

**7.1.4. Impact of Sensing Failure.** We now evaluate the impact of *sensing failure* on reconstruction performance. Sensing failure means that the robotic sensor platform cannot sample the field temporarily. In this set of simulations, we take the PE approach as an example and introduce random sensing failures. Specifically, each sensor has a sensing failure rate (10%, 20%, and 30%) in a sampling iteration. For sensors that experience sensing failure, their sampling positions will not be used in computing the drop of uncertainty  $\Omega$  and scheduling the swarm movement in the next iteration. We adopt the PE-MC and PE approaches with zero failure rate as baselines, in which the PE-MC approach gives the near-optimal swarm center regarding the PE metric. For each failure rate, we conduct six runs of simulations. The average information rewards are plotted in Figure 13. We can observe that our approach can achieve comparable reconstruction performance in the presence of relatively low sensing failure rate (e.g., 10%). As the sensing failure rate increases, the reconstruction performance drops. This is because the decreased sampling diversity leads to inaccuracy in swarm position selection. Note that, in addition to sensing failure, sensors are also subject to *hardware failure*, *motion*, and *control failure*. In particular, hardware failure means that the robotic sensor platform completely fails and the swarm will lose the failed node. Since a swarm consists of a limited number of nodes, the robotic sensor platform should be designed to have a low hardware failure rate to ensure long-term monitoring. Motion and control failure are caused by errors in swarm size control and sensor motion control such that the node cannot communicate with the swarm head. We have specifically presented two recovery mechanisms in Section 6.2 to address such failures.

**7.1.5. Effectiveness of Random Position Selection.** In this section, we analyze the effectiveness of random sensor position selection regarding information reward. In our approach, the position of each sensor in the next sampling iteration is randomly selected

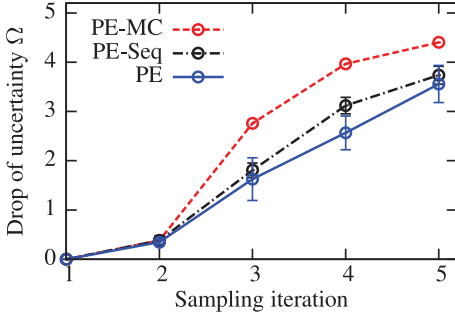


Fig. 14. Impact of sensor position selection on the information reward.

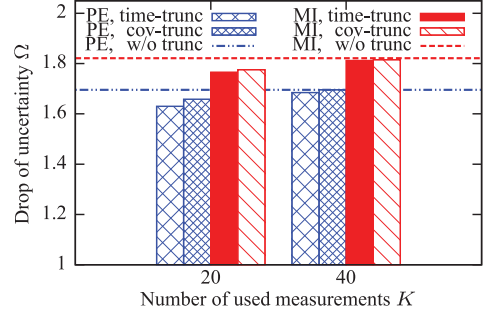


Fig. 15. Information reward in the 15<sup>th</sup> iteration versus the number of used measurements  $K$ .

by the swarm head within the rendezvous circle. We note that the proposed MI/PE-based metrics can also be used to guide the sensor position selection. Specifically, the swarm head sequentially selects each sensor's position based on the MI/PE-based metrics, such that each added sensor maximizes the information reward metric. We refer to the PE-based sequential sensor position selection approach as PE-Seq. This set of simulations only evaluate the PE-based approaches, which help us understand the impact of the random sensor position selection scheme. To make a fair comparison, we set the swarm radius of PE-Seq identical to that of PE in each iteration. Figure 14 plots  $\Omega$  versus the index of sampling iteration for PE-MC, PE-Seq, and PE, respectively. From the figure, we can see that the random sensor position selection (i.e., PE) gives comparable performance to the metric-guided sensor position selection (i.e., PE-Seq). The slightly better performance of PE-Seq is achieved at the cost of intensive computation overhead in determining each sensor's position. Due to the sequential execution process, this scheme cannot be executed on low-power sensing platforms at runtime. Therefore, within a rendezvous circle, the random placement of sensors does not significantly affect the information reward. Hence, our approach that combines metric-based swarm center position selection and in-swarm random sensor position selection not only is an effective solution in terms of information reward, but also simplifies the motion control of the sensor swarm as well as reduces the computation overhead of the swarm head.

**7.1.6. Impact of Historical Measurements Truncation.** In this set of simulations, we compare the performance of various combinations of the MI/PE-based metrics and the two truncation schemes presented in Section 5.4. The number of used historical measurements, namely,  $K$ , is set to be 20 or 40. The robotic sensor swarm is deployed in a  $1000 \times 1000 \text{ m}^2$  square region. Figure 15 plots the performance criterion  $\Omega$  at the 15<sup>th</sup> sampling iteration under various settings. Without the truncation scheme, all historical measurements are used and hence a greater  $\Omega$  is achieved. Moreover, we can see that  $\Omega$  increases with  $K$ . An interesting observation is that the truncation schemes with  $K = 40$  yield almost the same performance obtained by using all historical measurements. In addition, the *time-trunc* and *cov-trunc* schemes have comparable performance. Therefore, the *time-trunc* scheme with a small  $K$  can achieve satisfactory performance.

**7.1.7. Impact of Kernel Bandwidth.** The kernel bandwidths are important hyperparameters of the Gaussian process. We focus on the impact of the spatial kernel bandwidth  $\zeta_s$  while keeping the temporal kernel bandwidth  $\zeta_t$  fixed. Figure 16 plots performance criterion  $\Omega$  versus  $K$ , under various settings of  $\zeta_s$ . We can observe that  $\Omega$  increases with  $\zeta_s$ . This is because the stronger spatial correlation introduced by the larger  $\zeta_s$  can lead to a greater posterior entropy drop at the ungauged sites. Moreover, we can see

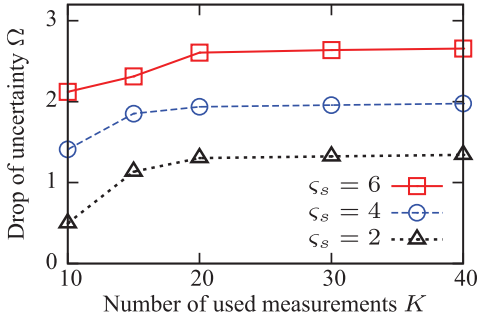


Fig. 16. Information reward versus the number of used measurements  $K$  under various  $\zeta_s$ .

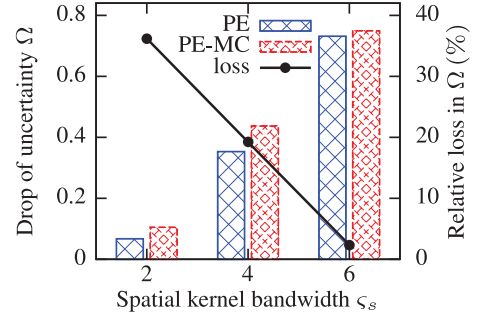


Fig. 17. Impact of kernel bandwidth on the approximation performance.

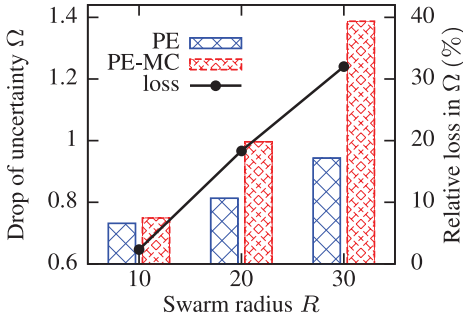


Fig. 18. Impact of swarm radius on the approximation performance.

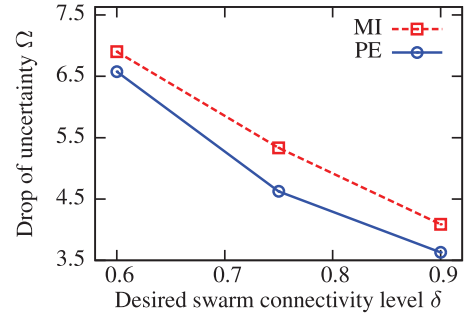


Fig. 19. Information reward versus desired swarm connectivity level.

from the figure that the performance becomes saturated after  $K$  is greater than 20 for various settings of  $\zeta_s$ .

**7.1.8. Approximation Performance.** In this section, by taking the PE approach as an example, we evaluate the performance of our heuristic that approximates the whole swarm by its center. As discussed in Section 7.1.3, the PE-MC approach gives the near-optimal swarm center regarding the PE metric. We assess the approximation performance in terms of the relative loss in  $\Omega$ , which is calculated as  $(\Omega_{\text{PE-MC}} - \Omega_{\text{PE}}) / \Omega_{\text{PE-MC}}$ . In the first set of simulations, we consider the impact of kernel bandwidth. Specifically, we vary the spatial kernel bandwidth  $\zeta_s$  while keeping the temporal kernel bandwidth  $\zeta_t$  and the swarm radius  $R$  fixed. The results are plotted in Figure 17. We can observe that relative loss decreases with kernel bandwidth. This result is consistent with the intuition that, when nearby positions are more correlated (i.e., a larger kernel bandwidth), the swarm center is a better representation of nearby positions. In the second set of simulations, we consider the impact of swarm radius. Specifically, we vary the swarm radius while keeping the kernel bandwidths fixed. Figure 18 shows the impact of swarm radius on the relative loss. From the figure, we can see that relative loss increases with swarm radius. This result is consistent with the intuition that the swarm center can better represent the whole swarm when the swarm size is smaller.

**7.1.9. Information Reward versus Swarm Connectivity.** As discussed in Section 4.3, there is a trade-off between the amount of information obtained by the sensor swarm and its connectivity level. In this set of simulations, we quantitatively evaluate the trade-off. Specifically, we vary the desired swarm connectivity level, that is,  $\delta$ , and compare the resulting information rewards. Other settings are consistent with those presented in

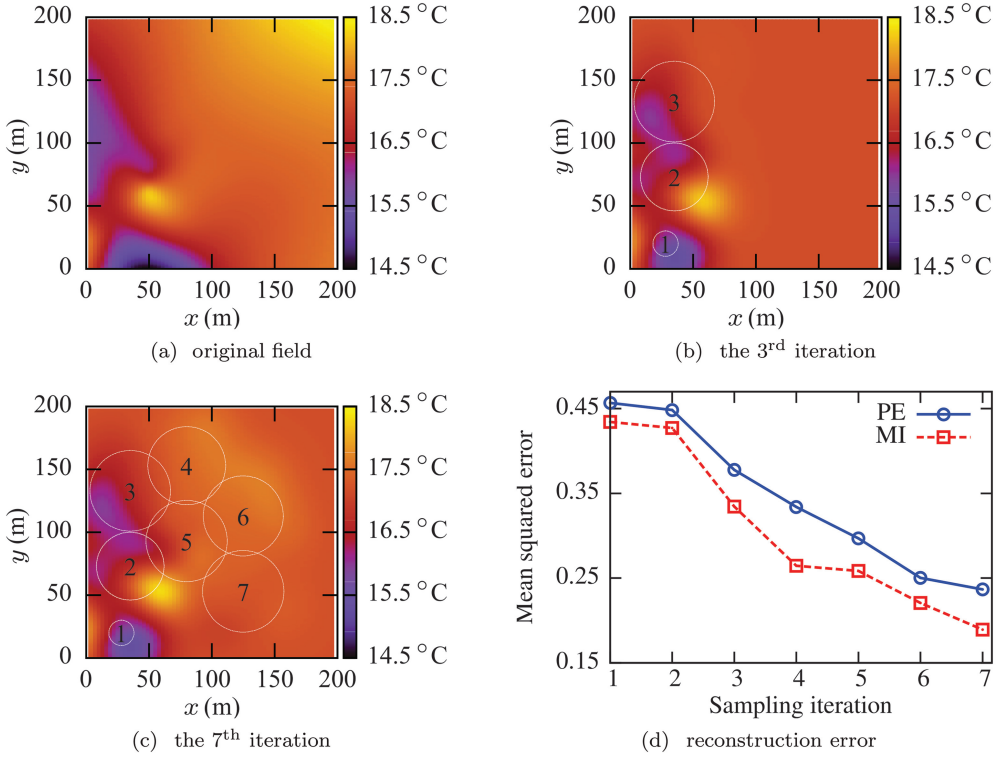


Fig. 20. Temperature field reconstruction using a robotic sensor swarm. The numbers in circles represent sequence of rendezvous circles.

Section 7.1.1. For each  $\delta$ , the control-theoretic algorithm adaptively tunes the radius of the rendezvous circle to maintain the swarm connectivity. According to Eq. (2), a larger  $\delta$  generally requires a smaller rendezvous circle, which will result in less information reward. Figure 19 plots the achieved information reward after five sampling iterations versus the desired swarm connectivity level. The decreasing relationship between drop of uncertainty and desired swarm connectivity level shown in Figure 19 verifies the trade-off.

**7.1.10. Accuracy of Field Reconstruction.** In this set of simulations, we reconstruct a field using 10 robotic sensors. We first generate a temperature field based on the temperature data [NAMOS Project 2006] collected at eight locations on the surface of Lake Fulmor, California, which has an area of about three acres. We have verified that temperature data follows the spatiotemporal Gaussian process in Section 5.1. However, the data at eight locations are not sufficient to drive the simulations. Therefore, we use an existing tool [Gaussian Surface Fit 2011] to fit a  $200 \times 200 \text{ m}^2$  ( $\approx 10$  acres) Gaussian process field based on the traces, as shown in Figure 20(a). For 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 [Xu et al. 2011; Krause et al. 2008]. In the reconstruction, the mean function  $\mathcal{M}(p, t)$  is set to be a fixed value of the average temperature in Figure 20(a). Figure 20(b) and Figure 20(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. Figure 20(d) plots



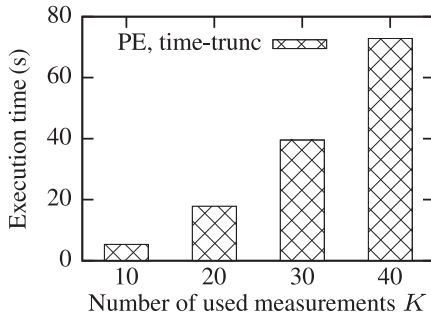


Fig. 21. Execution time of PE *time-trunc* scheduling versus the number of used measurements  $K$ .

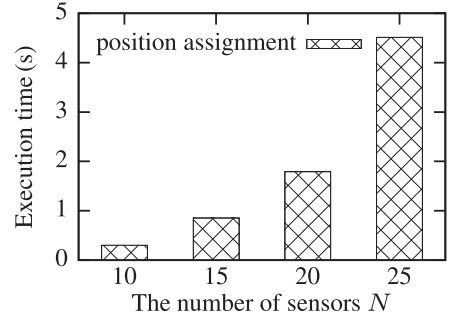


Fig. 22. Execution time of sensor position assignment versus the number of sensors  $N$ .

the mean squared error (MSE) of the reconstructed temperature field versus the index of sampling iteration. For comparison, we also include the MSE under MI approach. Due to space limitation, the corresponding reconstructed fields and swarm trajectories are omitted here. From Figure 20, we can see that the reconstruction accuracy is improved along with the movement of the swarm.

## 7.2. 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 a TelosB platform. We ported the C implementation of the Cholesky decomposition algorithm in GNU Scientific Library [GNU Scientific Library 2012] to TinyOS to invert the matrix in the swarm center selection algorithm. We also implement the Munkres algorithm in TinyOS to schedule each sensor's movement. Figure 21 and Figure 22 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 preceding 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 iteration can be much longer than that. Note that the MI metric and the *cov-trunc* scheme result in very long processing delays on the TelosB platform because of large search space and repeated matrix inversion operations. Therefore, they are only suitable for more powerful sensor platforms such as Imote2 [Imote2 2012].

## 8. CONCLUSION AND FUTURE WORK

In this article 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. We evaluate our approach by extensive trace-driven simulations and implementation on real sensor hardware. Our results show that the connectivity of robotic sensors can be maintained robustly in the presence of significant physical uncertainties. Moreover, despite the limited mobility, a

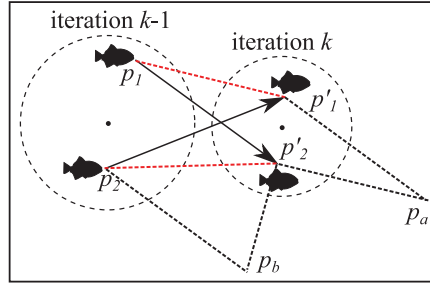


Fig. 23. Two crossing moving paths (i.e., segments  $\mathbf{p}_1\mathbf{p}'_2$  and  $\mathbf{p}_2\mathbf{p}'_1$ ) must not exist in the optimal solution to the element mapping problem.

sensor swarm can accurately reconstruct large, dynamic spatiotemporal aquatic fields, which validates the effectiveness of our information-theoretic movement scheduling algorithm. Lastly, our mechanisms incur low overhead on resource-constrained sensor motes.

In the future, we will implement our control- and information-theoretic algorithms on a testbed of motes. We will also evaluate the performance of our approach through emulation where the algorithms are executed on the motes, and the physical fields and sensor movements are simulated using real data traces. We will build fully functional robotic fish, and conduct real experiments in both a large indoor water tank and inland lakes for temperature field reconstruction. Moreover, we will extend our control-theoretic connectivity maintenance algorithm to multihop network topologies. We will also study how to integrate decentralized/distributed field reconstruction and sensor movement scheduling algorithms [Cortés 2009; Low et al. 2012; Chen et al. 2012] with our rendezvous-based swarm scheme.

## APPENDIXES

### A.1. Analysis of Controller's Stability and Convergence

This section presents the details of stability and convergence analysis in controller design. We first analyze the system reliability. The closed-loop transfer function (denoted by  $T_c(z)$ ) is given by  $T_c(z) = \frac{G_c(z)G_p(z)}{1+G_c(z)G_p(z)H(z)} = \frac{acz}{z-(\beta-\alpha c)}$ , which has a pole at  $z = \beta - \alpha c$ . The closed-loop system is stable if this pole is located within the unit circle [Ogata 1995], that is,  $|\beta - \alpha c| < 1$ . Therefore, the sufficient condition for stability is  $\frac{\beta-1}{c} < \alpha < \frac{\beta+1}{c}$ . We then analyze the steady-state error of the connectivity control system. The open-loop transfer function (denoted by  $T_o(z)$ ) is given by  $T_o(z) = G_c(z)G_p(z)H(z) = \frac{ac}{z-\beta}$ . By letting  $\beta = 1$ , the system is a type-1 system [Ogata 1995] that exhibits no steady-state error in response to step inputs. Therefore, the condition for both stability and convergence is  $\beta = 1$  and  $0 < \alpha < \frac{2}{c}$ .

### A.2. Proof of Collision-Free Trajectories

In this section, by contradiction, we prove that there is no crossing among sensors' moving paths under the movement scheduling scheme discussed in Section 5.5. We assume that there are crossings among sensors' moving paths. Without loss of generality, we consider any two crossing moving paths, as depicted by segments  $\mathbf{p}_1\mathbf{p}'_2$  and  $\mathbf{p}_2\mathbf{p}'_1$  in Figure 23. We now prove that if the two sensors move along segments  $\mathbf{p}_1\mathbf{p}'_1$  and  $\mathbf{p}_2\mathbf{p}'_2$  without crossing, the total movement distance can be reduced. Therefore, a movement schedule that contains any crossing must not be the optimal solution to the element mapping problem.



A necessary condition for the intersecting of segments  $\mathbf{p}_1\mathbf{p}'_2$  and  $\mathbf{p}_2\mathbf{p}'_1$  is that points  $p_1$  and  $p'_2$  are located at different sides of segment  $\mathbf{p}_2\mathbf{p}'_1$ , and so are points  $p_2$  and  $p'_1$  with segment  $\mathbf{p}_1\mathbf{p}'_2$ . Let  $p_a$  denote a point in the aquatic field such that the quadrilateral  $p_1p'_2p_ap'_1$  forms a parallelogram. So is  $p_b$  that gives parallelogram  $p_1p_2p_bp'_2$ . Therefore, the sum of sensors' movement distances with crossing trajectories is given by  $\|\mathbf{p}_2\mathbf{p}'_1\|_{\ell_2} + \|\mathbf{p}'_1\mathbf{p}_a\|_{\ell_2}$ , and that without crossing trajectories is given by  $\|\mathbf{p}_2\mathbf{p}'_2\|_{\ell_2} + \|\mathbf{p}'_2\mathbf{p}_a\|_{\ell_2}$ . Hence, the quadrilateral  $p_2p_bp_ap'_1$  is a parallelogram as well. Since point  $p'_2$  locates inside of parallelogram  $p_2p_bp_ap'_1$ , by geometry, the sum of distances it is to the across corners (i.e.,  $p_2$  and  $p_a$ ) is smaller than that of the corner  $p'_1$ , namely  $\|\mathbf{p}_2\mathbf{p}'_2\|_{\ell_2} + \|\mathbf{p}'_2\mathbf{p}_a\|_{\ell_2} < \|\mathbf{p}_2\mathbf{p}'_1\|_{\ell_2} + \|\mathbf{p}'_1\mathbf{p}_a\|_{\ell_2}$ .

## REFERENCES

- T. Adbelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu. 2008. Introduction to control theory and its application to computing systems. In *Performance Modeling and Engineering*, Springer, 185–215.
- J. S. Baras, X. Tan, and P. Hovareshti. 2003. Decentralized control of autonomous vehicles. In *Proceedings of the 42<sup>nd</sup> IEEE Conference on Decision and Control (CDC'03)*. 1532–1537.
- R. E. Burkard, M. Dell'amico, and S. Martello. 2009. *Assignment Problems*. Society for Industrial Mathematics.
- X. Chang, R. Tan, G. Xing, Z. Yuan, C. Lu, Y. Chen, and Y. Yang. 2011. Sensor placement algorithms for fusion-based surveillance networks. *IEEE Trans. Parallel Distrib. Syst.* 22, 8, 1407–1414.
- J. Chen, K. H. Low, C. K. Y. Tan, A. Oran, P. Jaillet, J. M. Dolan, and G. S. Sukhatme. 2012. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proceedings of the 28<sup>th</sup> Conference on Uncertainty in Artificial Intelligence*. 163–173.
- Y. Chen and A. Terzis. 2011. On the implications of the log-normal path loss model: An efficient method to deploy and move sensor nodes. In *Proceedings of the 9<sup>th</sup> ACM Conference on Embedded Networked Sensor Systems (SenSys'11)*. ACM Press, New York, 26–39.
- J. Cortes. 2009. Distributed kriged kalman filter for spatial estimation. *IEEE Trans. Autom. Control* 54, 12, 2816–2827.
- K. Dantu, B. Kate, J. Waterman, P. Bailis, and M. Welsh. 2011. Programming micro-aerial vehicle swarms with karma. In *Proceedings of the 9<sup>th</sup> ACM Conference on Embedded Networked Sensor Systems (SenSys'11)*. ACM Press, New York, 121–134.
- M. C. De Gennaro and A. Jadbabaie. 2006. Decentralized control of connectivity for multi-agent systems. In *Proceedings of the 45<sup>th</sup> IEEE Conference on Decision and Control (CDC'96)*. 3628–3633.
- C. Detweiler, M. Doniec, M. Jiang, M. Schwager, R. Chen, and D. Rus. 2010. Adaptive decentralized control of underwater sensor networks for modeling underwater phenomena. In *Proceedings of the 8<sup>th</sup> ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*. ACM Press, New York, 253–266.
- J. M. Dolan, G. Podnar, S. Stancliff, E. Lin, J. Hosler, T. Ames, J. Moisan, T. Moisan, J. Higinbotham, and A. Elfes. 2007. Harmful algal bloom characterization via the telesupervised adaptive ocean sensor fleet. In *NASA Science Technology Conference*.
- Gaussian Surface Fit. 2011. <http://mathworks.com/matlabcentral/fileexchange/31485>.
- Gnu Scientific Library. 2012. <http://gnu.org/software/gsl>.
- Habs and Lake Mendota. 2012. <http://blooms.uwcf.org/mendota>.
- T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. 2003. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23<sup>rd</sup> International Conference on Distributed Computing Systems (ICDCS'03)*. IEEE, 46–55.
- Imote2 Datasheet. 2012. Crossbow technology. <http://www.xbow.com/>.
- G. Judd, X. Wang, and P. Steenkiste. 2008. Efficient channel-aware rate adaptation in dynamic environments. In *Proceedings of the 6<sup>th</sup> International Conference on Mobile Systems, Applications and Services (MobiSys'08)*. ACM Press, New York, 118–131.
- A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. 2006. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5<sup>th</sup> International Conference on Information Processing in Sensor Networks (IPSN'06)*. 2–10.
- A. Krause, A. Singh, and C. Guestrin. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.* 9, 235–284.

- S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. 2006. ATPC: Adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4<sup>th</sup> International Conference on Embedded Networked Sensor Systems (SenSys'06)*. ACM Press, New York, 223–236.
- H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu. 2010. Automatic and robust breadcrumb system deployment for indoor firefighter applications. In *Proceedings of the 8<sup>th</sup> International Conference on Mobile Systems, Applications and Services (MobiSys'10)*. ACM Press, New York, 21–34.
- K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson. 2012. Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In *Proceedings of the 11<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems*. 105–112.
- K. H. Low, J. M. Dolan, and P. Khosla. 2008. Adaptive multi-robot wide-area exploration and mapping. In *Proceedings of the 7<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems*. 23–30.
- K. H. Low, J. M. Dolan, and P. Khosla. 2009. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proceedings of the 19<sup>th</sup> International Conference on Automated Planning and Scheduling*. 233–240.
- K. H. Low, J. M. Dolan, and P. Khosla. 2011. Active markov information-theoretic path planning for robotic environmental sensing. In *Proceedings of the 10<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems*. 753–760.
- R. Maheshwari, S. Jain, and S. R. Das. 2008. A measurement study of interference modeling and scheduling in low-power wireless networks. In *Proceedings of the 6<sup>th</sup> ACM Conference on Embedded Networked Sensor Systems (SenSys'08)*. ACM Press, New York, 141–154.
- NAMOS Project. 2006. <http://robotics.usc.edu/~namos/>.
- K. Ogata. 1995. *Discrete-Time Control Systems*. Prentice Hall.
- A. Purohit, Z. Sun, F. Mokaya, and P. Zhang. 2011. SensorFly: Controlled-mobile sensing platform for indoor emergency response applications. In *Proceedings of the 10<sup>th</sup> International Conference on Information Processing in Sensor Networks (IPSN'11)*. 223–234.
- L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. 2007. A general model of wireless interference. In *Proceedings of the 13<sup>th</sup> Annual International Conference on Mobile Computing and Networking (MobiCom'07)*. ACM Press, New York, 171–182.
- K. M. Ramachandran and C. P. Tsokos. 2009. *Mathematical Statistics with Applications*. Academic Press.
- T. S. Rappaport. 1996. *Wireless Communications: Principles and Practice*. Prentice Hall.
- C. E. Rasmussen. 2006. *Gaussian Processes for Machine Learning*. MIT Press.
- 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. 2007. A wireless internet-based observatory: The real-time coastal observation network (recon). In *Proceedings of the Marine Technology Society/IEEE Oceans Conference*.
- D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry. 2004. Underwater gliders for ocean research. *Marine Technol. Soc. J.* 38, 2, 73–84.
- Science Daily. 2004. Researchers experiment with solar underwater robots. <http://sciencedaily.com/releases/2004/12/041212081548.htm>.
- A. Singh, A. Krause, C. Guestrin, and W. Kaiser. 2009. Efficient informative sensing using multiple robots. *J. Artif. Intell. Res.* 34, 1, 707–755.
- A. Singh, R. Nowak, and P. Ramanathan. 2006. Active learning for adaptive mobile sensing networks. In *Proceedings of the 5<sup>th</sup> International Conference on Information Processing in Sensor Networks (IPSN'06)*. 60–68.
- X. Tan. 2011. Autonomous robotic fish as mobile sensor platforms: Challenges and potential solutions. *Marine Technol. Soc. J.* 45, 4, 31–40.
- J. N. Twigg, J. R. Fink, P. L. Yu, and B. M. Sadler. 2012. RSS gradient-assisted frontier exploration and radio source localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'12)*. 889–895.
- H. Wang, K. Yao, G. Pottie, and D. Estrin. 2004. Entropy-based sensor selection heuristic for target localization. In *Proceedings of the 3<sup>rd</sup> International Conference on Information Processing in Sensor Networks (IPSN'04)*. 36–45.
- Y. Wang, R. Tan, G. Xing, J. Wang, and X. Tan. 2012. Accuracy-aware aquatic diffusion process profiling using robotic sensor networks. In *Proceedings of the 11<sup>th</sup> International Conference on Information Processing in Sensor Networks (IPSN'12)*. 281–292.
- Y. Xu, J. Choi, and S. Oh. 2011. Mobile sensor network navigation using gaussian processes with truncated observations. *IEEE Trans. Robot.* 27, 6, 1118–1131.

- B. Zhang and G. S. Sukhatme. 2007. Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'07)*. 3673–3680.
- M. Zuniga and B. Krishnamachari. 2004. Analyzing the transitional region in low power wireless links. In *Proceedings of the 1<sup>st</sup> IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*. 517–526.

Received September 2012; revised July 2013; accepted July 2013