12-2017

# Energy-aware Sparse Sensing of Spatial-temporally Correlated Random Fields

Zuoen Wang
*University of Arkansas, Fayetteville*

Energy-aware Sparse Sensing of Spatial-temporally
Correlated Random Fields

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering with a concentration in Electrical Engineering

by

Zuoen Wang
Huazhong University of Science and Technology, China
Bachelor of Science in Electrical Engineering, 2010
Harbin Institute of Technology, China
Master of Science in Electrical Engineering, 2012

December 2017
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

---

Dr. Jingxian Wu
Dissertation Director

---

Dr. Roy A. McCann
Committee Member

---

Dr. Yue Zhao
Committee Member

---

Dr. Qinghua Li
Committee Member

**Abstract**

This dissertation focuses on the development of theories and practices of energy aware sparse sensing schemes of random fields that are correlated in the space and/or time domains. The objective of sparse sensing is to reduce the number of sensing samples in the space and/or time domains, thus reduce the energy consumption and complexity of the sensing system. Both centralized and decentralized sensing schemes are considered in this dissertation. Firstly we study the problem of energy efficient Level set estimation (LSE) of random fields correlated in time and/or space under a total power constraint. We consider uniform sampling schemes of a sensing system with a single sensor and a linear sensor network with sensors distributed uniformly on a line where sensors employ a fixed sampling rate to minimize the LSE error probability in the long term. The exact analytical cost functions and their respective upper bounds of these sampling schemes are developed by using an optimum thresholding-based LSE algorithm. The design parameters of these sampling schemes are optimized by minimizing their respective cost functions. With the analytical results, we can identify the optimum sampling period and/or node distance that can minimize the LSE error probability. Secondly we propose active sparse sensing schemes with LSE of a spatial-temporally correlated random field by using a limited number of spatially distributed sensors. In these schemes a central controller is designed to dynamically select a limited number of sensing locations according to the information revealed from past measurements,and the objective is to minimize the expected level set estimation error. The expected estimation error probability is explicitly expressed as a function of the selected sensing locations, and

the results are used to formulate the optimal sensing location selection problem as a combinatorial problem. Two low complexity greedy algorithms are developed by using analytical upper bounds of the expected estimation error probability. Lastly we study the distributed estimations of a spatially correlated random field with decentralized wireless sensor networks (WSNs). We propose a distributed iterative estimation algorithm that defines the procedures for both information propagation and local estimation in each iteration. The key parameters of the algorithm, including an edge weight matrix and a sample weight matrix, are designed by following the asymptotically optimum criteria. It is shown that the asymptotically optimum performance can be achieved by distributively projecting the measurement samples into a subspace related to the covariance matrices of data and noise samples.

**Acknowledgments**

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Jingxian Wu, a tremendous mentor for me, for the continuous guidance and support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His advice and guidance on all the time of my research research as well as on my career have been invaluable for me.

Besides my supervisor, I would like to thank the rest of my thesis committee members Dr. Roy McCann , Dr.Yue Zhao , and Dr. Qinghua Li. for serving as my committee members. Additionally I would also like to express my special appreciation for my former committee member Dr. Jing Yang for her insightful comments and encouragement to widen my research from various perspectives.

Also my thanks extend to all the present and former members of the Intelligent Information Sensing, Processing and Transmission Lab, including Ning Sun, Guoqing Zhou, Gang Wang, Qing Guo, Shanqi Sun, Boyu Wang, Xianwen Wu, Israel Z. Akingeneye, Samrat Nath, Md Abul Hayat, Tanny Chavez Esparza, Yaze Li and so on and visiting students and scholars including Dr. Guodong Li, Dr. Yinjing Guo, Dr. Yingli Cao, Dr. Yutao Ma, Dr. Changhua Zhao, Dali Hu, Jing Zhao, Jian Li, Weixi Zhou and so on, for all the happy or hard time we have had in the last four years. I am also grateful to my friends both in the United States and China who have supported me along the way.

Last but not least, I would like to thank my parents, parents-in-law, brothers, wife, and little girl Elisa. They have been always there supporting me and encouraging me. The warmth of the family love helps me to overcome all the difficulties, not only in the study but

also in the daily life.

## Contents

## List of Figures

## List of Papers

- **Chapter 2**, Zuoen Wang, Jingxian Wu, and Jing Yang, "Energy Efficient Wireless Sensing for Level Set Estimations," *IEEE Access*, vol. 3, pp. 1480-1490, 2015.

- **Chapter 3**, Zuoen Wang, Jingxian Wu, Jing Yang, and Hai Lin, "Optimal Energy Efficient Level Set Estimation of Spatially-Temporally Correlated Random Fields," in *Communications (ICC), 2016 IEEE International Conference on*, pp. 1-6., June 2016.

- **Chapter 4**, Zuoen Wang, Jing Yang, and Jingxian Wu, "Level Set Estimation of Spatial-temporally Correlated Random Fields with Active Sparse Sensing," *IEEE Transactions on Aerospace and Electronic Systems.*, vol. 53, pp. 862-876, Feb. 2017.

- **Chapter 5**, Zuoen Wang, Jingxian Wu, Jing Yang, and Yingli Cao, "Optimum Distributed Estimation of a Spatially Correlated Random Field," submitted to *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

# Chapter 1

# Introduction

## 1.1 Background and Motivation

Wireless sensor networks (WSNs) sometimes called wireless sensor and actuator networks (WSANs) [1] have experienced significant growth over the past decade A WSN consists of hundreds of low-cost and low-power autonomous sensors with limited sensing, computation, and wireless communication capabilities. Sensors in a WSN are spatially distributed over a target area to continuously observe physical phenomena, such as air pressure, vibration, pressure, temperature, aggregated power level of wireless signals, density of toxic gases, etc..

WSNs have been widely used in many scientific and engineering applications, including search and rescue, precision agriculture, industrial monitoring, disaster relief, spectrum sensing, landslide detection, forest fire detection, water/waste/air pollution monitoring, natural disaster prevention, etc.

Data taken by sensors among all these application scenarios share an intrinsic and significant characteristic of spatial and temporal correlations due to the nature of the energy-radiating physical phenomenon. Typically sensors are deployed spatially in high density in WSN fields to cover target areas with the aim to reliably detect and estimate a certain event of interest. Thus multiple observations from among close sites usually share correlated information about the event. Similarly some of WSN applications may require sensors to keep

tracking of a specific event. Each individual sensor is activated to take sequential observations in a fixed period or at actively selected instants. These sequential observations from the same sensor may also contain temporally correlated information on an event of interest in a WSN field.

Even though these spatial and temporal correlated observations can be exploited to improve detection and estimation accuracy on the signal of event of interest, the problem on how to reduce or limit redundant observations taken by sensors while satisfying given detection and estimation accuracy remains challenging and attractive. This problem is referred as *sparse sensing* in this dissertation. The objective of sparse sensing is to reduce the number of space- and/or time-domain samples required by the sensing application. Reducing the number of sensing samples can reduce the energy consumption and complexity of the sensing system.

In many WSN applications estimating or identifying level sets of an event of interest is the primary task, while estimating the values of original function of the event away from the level set boundary is often secondary if not irrelevant. Level sets of an event in spatial-temporally correlated field are defined as regions where function values of the event exceed a certain threshold. Level set estimation (LSE) is the process of using observations of function of an event to estimate the region(s) where the function value exceeds some critical threshold. Estimating level set can be equivalently considered as a mapping problem that draws the level contour or boundary in the field. The problem of sparse sensing can be naturally combined with level set estimation. Intuitively data that are further away from the boundary are usually quite distinct from the level of interests, thus there is less ambiguity in terms of level set identification in those regions. Therefore it is desirable to collect less data samples or

place less sensors at the locations where the boundary is not likely to lie.

WSNs can be classified into two categories, centralized and decentralized WSNs. In a centralized WSN, sensors are usually of lower cost and low complexity because the number of sensor nodes deployed in a target area may be in the order of hundreds or thousands. The measurement results from all sensors are congregated at a fusion center (FC) with much higher capabilities of processing and computing, either through direct transmission or by using other nodes as relays. Information processing is performed centrally at the FC. Centralized WSNs are easy to design, but they also suffer from a lot of limitations such as high cost of the FC, communication bottlenecks at areas close to the FC, susceptible to node failures, etc. These problems can be easily addressed by a decentralized WSN, where information processing is performed at each sensor node in a distributed manner without the need of a central controller. Information processing is performed collaboratively among nodes through iterative information exchange among neighboring nodes.

Distributed estimation is one of the most fundamental collaborative information processing problems in distributed WSNs, where the nodes distributively perform estimation of certain physical quantities through information exchange [2–9]. Most distributed estimations involve two components: a local estimator and a distributed consensus algorithm that can be used to improve estimation performance.

## 1.2 Objectives

Our research in this dissertation is dealing with sparse sensing problem and aimed at the methods to exploit spatial and temporal correlation ingrained in random fields of WSNs to enhance system performance with limited energy or sensor resources. Both centralized and

decentralized networks are studied in this dissertation.

Firstly we study the ways to utilize the temporal or/and spatial correlation to improve accuracy of level set estimations under a total power constraint. An accurate LSE usually demands a large amount of data to be collected, processed, and transmitted, and energy is consumed during the sensing and transmission of each data sample. Limited energy supplies in wireless sensing systems may not be able to meet the high energy demands imposed by the large amount of data. Under a total power constraint, more data samples in a unit time or area result in less energy per sample at the FC, and this will negatively affect the estimation performance. On the other hand, a smaller sampling interval in time or space means a stronger correlation among the samples, which may positively contribute to the estimation accuracy. Therefore there is a fundamental tradeoff between the amount of sensing data and energy supply. It is critical to identify the optimum sensing scheme that can balance this tradeoff, such that we can significantly reduce the amount of data to be collected and still achieve an accurate LSE under a stringent power constraint.

In particular we at the beginning study the problem of energy efficient LSE of a time-varying random field under a total power constraint. In this simple case the fusion center of a wireless sensing system performs LSE by using discrete-time samples collected by a sensor. The sampling period has to be optimized to minimize the estimation error and balance the data-energy tradeoff. We extend from the results of this simple case, and investigate the problem of optimum energy efficient LSE of random fields correlated in both time and line (1-D) space. Both node distance and sampling period are required to be optimized simultaneously in this case to guarantee best possible estimation accuracy.

Secondly we develop active sparse sensing scheme for LSE of a spatial-temporally correlated random field by using a limited number of spatially distributed sensors. In this scheme a central controller is designed to dynamically select a limited number of sensing locations according to the information revealed from past measurements, and the objective is to minimize the expected level set estimation error.

Lastly, we focus on the problem of distributed estimation of a spatially correlated random field with decentralized WSNs. Sensor nodes in the network take spatial samples of the random field, then each node estimates the values of arbitrary points on the random field by iteratively exchanging information with each other without the need of a central controller. The objective is to minimize the estimation mean squared error (MSE) while ensuring all nodes reach a distributed consensus on the estimation results.

The study of theories and practices of sparse sensing are incarnated throughout this dissertation by means of practical design, theoretical analysis and extensive simulations under various system configurations.

## 1.3 Dissertation Outline

In the rest of this dissertation, four chapters are used to summarize what we have done so far about sparse sensing of spatial-temporally correlated random fields in WSNs. For the specific topic in each chapter, we give the background introduction, literature review, proposed methodology, and results of simulations and/or experiments. Finally, we use the last chapter to conclude contributions of this dissertation and discuss the future works . The outline of the rest of the dissertation is listed as follows.

*Chapter 2*: In this chapter, we study energy efficient LSE of a time-varying random field under a total power constraint. The fusion center of a wireless sensing system performs LSE by using discrete-time samples collected by a sensor. Two sampling schemes are considered in this chapter: a dynamic active sampling scheme that sequentially and adaptively selects the next sampling instant in a myopic manner with knowledge learned from previous samples, and a uniform sampling scheme that employs a fixed sampling rate to minimize the LSE error probability in the long term.

*Chapter 3*: We investigate optimum LSE of a correlated random field in both time and line space under a total power constraint. A linear sensor network is used to take discrete samples of a spatial-temporally correlated random field, and the sensors operate with limited power supply. The samples are congregated at a fusion center, which performs LSE of the random field. Under the Gaussian process (GP) framework, we first develop an optimum LSE algorithm that can minimize the LSE error probability. The results are then used to derive the exact LSE error probability with the assistance of frequency domain analysis. With the analytical results, we can identify the optimum node distance and sampling period that can minimize the LSE error probability.

*Chapter 4*: We propose active sparse sensing schemes with LSE of a spatial-temporally correlated random field by using a limited number of spatially distributed sensors. In these schemes a central controller is designed to dynamically select a limited number of sensing locations according to the information revealed from past measurements, with the objective to minimize the expected level set estimation error. The expected estimation error probability is explicitly expressed as a function of the selected sensing locations, and the results are used to formulate the optimal sensing location selection problem as a combinatorial problem.

Two low complexity greedy algorithms are developed by using analytical upper bounds of the expected estimation error probability. Both simulation and experiment results demonstrate that the greedy algorithms can achieve significant performance gains over baseline passive sensing algorithms and the GP Upper Confidence Bound (GP-UCB) level set estimation algorithm.

*Chapter 5*: In this chapter we study the distributed estimations of a spatially correlated random field with decentralized WSNs. Nodes in the network take spatial samples of the random field, then each node estimates the values of arbitrary points on the random field by iteratively exchanging information with each other without need of a central controller. The objective is to minimize the estimation mean squared error (MSE) while ensuring all nodes reach a distributed consensus on the estimation results. Specifically, we propose a distributed iterative estimation algorithm that defines the procedures for both information propagation and local estimation in each iteration. The key parameters of the algorithm, including an edge weight matrix and a sample weight matrix, are designed by following the asymptotically optimum criteria. It is shown that the asymptotically optimum performance can be achieved by distributively projecting the measurement samples into a subspace related to the covariance matrices of data and noise samples. Simulation and experimental results show that all nodes in a large network can obtain accurate estimation results with much less iterations than existing algorithms.

*Chapter 6*: Conclusion remarks are drawn in this chapter. The major contributions of this research proposal is summarized, and future work is discussed.

### 1.4 References


[1] Ian F Akyildiz and Ismail H Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad hoc networks*, 2(4):351–367, 2004.

[2] Sergio Barbarossa and Gesualdo Scutari. Bio-inspired sensor network design. *IEEE Signal Processing Magazine*, 24(3):26–35, 2007.

[3] Alexander Bertrand and Marc Moonen. Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks. *IEEE Transactions on Signal Processing*, 59(5):2320–2330, 2011.

[4] Federico S Cattivelli, Cassio G Lopes, and Ali H Sayed. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, 56(5):1865–1877, 2008.

[5] Jorge Cortés. Distributed kriged kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827, 2009.

[6] Alexandros G Dimakis, Soummya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[7] Lin Li, Anna Scaglione, and Jonathan H Manton. Distributed principal subspace estimation in wireless sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):725–738, 2011.

[8] Ioannis D Schizas, Georgios B Giannakis, Stergios I Roumeliotis, and Alejandro Ribeiro. Consensus in ad hoc wsns with noisy linkspart ii: Distributed estimation and smoothing of random signals. *IEEE Transactions on Signal Processing*, 56(4):1650–1666, 2008.

[9] Ioannis D Schizas, Alejandro Ribeiro, and Georgios B Giannakis. Consensus in ad hoc wsns with noisy linkspart i: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350–364, 2008.

# Chapter 2

## Energy efficient wireless sensing for level set estimations

### 2.1 Abstract

Level set estimation (LSE) is the process of using noisy observations of an unknown function to estimate the region(s) where the function values lie above a given threshold. It has a wide range of applications in many scientific and engineering areas, such as spectrum sensing or environment monitoring. In this paper, we study the energy efficient LSE of a time-varying random field under a total power constraint. The fusion center (FC) of a wireless sensing system performs LSE by using discrete-time samples collected by a sensor. An accurate LSE usually requires a large number of samples to be collected and transmitted. However, most wireless sensing systems operate with a stringent power constraint that may not be able to meet the high energy demands imposed by the large amount of data. The gap between energy demands and supplies is a direct result of the so-called "big data" problem. It is critical to develop energy efficient sampling schemes that can bridge this gap by reducing the amount of data required by LSE. Two sampling schemes are considered in this paper: a dynamic active sampling scheme that sequentially and adaptively selects the next sampling instant in a myopic manner with knowledge learned from previous samples, and a uniform sampling scheme that employs a fixed sampling rate to minimize the LSE error probability in the long term. The exact analytical cost functions and their respective

upper bounds of both sampling schemes are developed by using an optimum thresholding-based LSE algorithm. The design parameters of both sampling schemes are optimized by minimizing their respective cost functions. Analytical and simulation results demonstrate that both sampling schemes can significantly reduce the amount of data collected by the system while obtain accurate LSE under a stringent power constraint. In addition, the uniform sampling scheme slightly outperforms the dynamic active sampling scheme.

## 2.2 Introduction

Level set estimation (LSE) is the process of using noisy observations of an unknown function defined on a Hilbert space to estimate the region(s) where the function amplitude lies above a given threshold. It has a wide range of applications in many scientific and engineering areas. For example, the objective of spectrum sensing in cognitive radio networks is to identify the boundary of "spectrum holes" in the space, time, and frequency domains [1]. Other applications include the monitoring of the contours of pollution, sunlight, temperature, or rainfalls for biosystem ecology tracking [10, 11], etc. In these and many other applications, identifying level sets is the primary task, while estimating the value of the function away from the level set boundary is often secondary, if not irrelevant. Consequently, level set estimation can be equivalently considered as a mapping problem that draws the level contour or boundary in a random field.

LSE can be performed by applying standard binary classifications to the implicit function using probability models [6, 9, 12]. The binary classification approach ignores the difference between the actual function value and the threshold, and such information contains salient information that can improve the LSE accuracy. Another popular approach is to estimate

the values of the underlying functions through regression, and then obtain the level set by thresholding the estimated function values at the critical value [4, 14].

Most LSE methods are applicable in a static setting, that is, the measurements in the field are given or passively provided [4,16,18]. Recently, it has been proposed to dynamically adjust the sensing strategy based on past sensing results [2,7,20] by using active learning [15]. The dynamic LSE employs sequential decision makings, and it can accurately track the level set in a time-varying random field. None of the above mentioned works consider the constraints imposed by the limited energy supply, which is one of the main performance limiting factors in wireless sensing systems.

An accurate LSE usually demands a large amount of data to be collected, processed, and transmitted, and energy is consumed during the sensing and transmission of each data sample. However, most low power wireless sensors are equipped with extremely limited energy supplies such as small batteries or energy harvesting devices. The limited energy supplies in wireless sensing systems may not be able to meet the high energy demands imposed by the large amount of data. The big gap between energy supplies and demands is a direct result of the so-called "big data" problem, and it imposes formidable challenges for system designs. Under a total power constraint, more data samples in a unit time results in less energy per sample, or a lower signal-to-noise ratio (SNR) per sample at the FC, and this will negatively affect the estimation performance. On the other hand, a higher sampling rate means a stronger temporal correlation among the samples, which may positively contribute to the estimation accuracy [19]. Therefore there is a fundamental tradeoff between the amount of sensing data and energy supply. It is critical to identify the optimum sensing scheme that can balance this tradeoff, such that we can significantly reduce the amount of

data to be collected and still achieve an accurate LSE under a stringent power constraint.

In this paper, we study the optimum LSE in a power-constrained wireless sensing system by explicitly identifying and optimizing the data-energy tradeoff. A wireless sensor samples the time-varying random event and transmits the discrete-time samples to a fusion center (FC), which performs LSE by using distorted observations of the discrete-time samples. The optimum sampling of a power-constrained wireless sensing system that can minimize the estimation mean squared error (MSE) of a random field has been studied in [17, 19]. The LSE problem studied in this paper is different from [17, 19], in that our objective is not to reconstruct the entire function, but to estimate the level set of the underlying function.

We introduce a Gaussian process (GP) prior model to capture the temporal correlation inherent in the random field [13]. Under the GP framework, we first show that the time-averaged LSE error can be achieved by performing a GP regression with all discrete-time data samples and then thresholding the regression results. With the thresholding-based LSE method, two sampling schemes are considered in this paper: a dynamic active sampling scheme that sequentially and adaptively selects the next sampling instant by using knowledge learned from previous samples, and a uniform sampling scheme that employs a fixed sampling rate. For uniform sampling with a given sampling rate, we know the exact sampling instants for all future samples, thus the uniform sampling scheme can minimize the LSE error probability averaged over the entire time duration. On the other hand, dynamic active sampling needs to select the sampling instants sequentially, thus the optimization needs to be performed in a myopic manner, that is, the cost function is the LSE error probability averaged in time up to the next possible sampling instant.

Exact analytical cost functions and their respective upper bounds of both sampling

schemes are developed by following the GP prior model, and they are expressed as functions as various system parameters, such as the hardware energy consumption, the power constraint of the sensor, the probability distribution of the random field, and the sampling instants, etc. The optimum sampling schemes are designed by minimizing their respective cost functions. Numerical and simulation results demonstrate that both sampling schemes can balance the data-energy tradeoff by significantly reducing the amount of required data while still achieving accurate LSE, and the uniform sampling scheme slightly outperforms dynamic active sampling.

The remainder of this paper is organized as follows. The system model and problem formulation are given in Section II. Section III presents an optimum LSE estimation algorithm if the sampling scheme has been selected. The dynamic active sampling and uniform sampling are developed in Sections IV and V, respectively. Numerical and simulation results are presented in Section VI, and Section VII concludes the paper.

### 2.3 System Model

We consider the sensing and monitoring of the level set of a time-varying random event, $x(t)$, where $t$ is the time variable. The random event can be used to model temperature, air pressure, or density of toxic gases, etc.

*Assumptions 2.1*: We make the following assumptions about the random event $x(t)$:

1) The prior distribution of $\{x(t)\}$ is a zero-mean Gaussian process (GP) with covariance function $k(t,t') = \mathbb{E}[x(t)x(t')]$, i.e, $x \sim GP(0, k(t,t'))$.

2) The Gaussian process is wide sense stationary in time, and

$$k(t,t') = \rho^{|t-t'|}, \tag{2.1}$$

where $\rho \in [0,1]$ is the temporal correlation coefficient.

At time $t$ , we are interested in identifying the $\gamma$-level set of $\{x(t)\}$, which is define as

$$\mathcal{S}(t) := \{t' \in [0,t] : x(t') > \gamma\} \tag{2.2}$$

We assume $\gamma > 0$ without loss of generality.

The level set will be estimated by using distorted observations of the random event. Due to energy limit, the sensing system can only take discrete-time samples of the continuous-time random event. The collected discrete-time samples are transmitted to a fusion center (FC). Denote the sampling instants as $t_i$, for $i = 1, 2, \cdots$. It is assumed that the $i$-th sampling operation consumes an energy of $E_{0i} = E_c + E_i$ Joul, where the constant $E_c$ is due to hardware power consumption, and $E_i$ is the transmission energy of the $i$-th sample. The samples observed at the FC can be represented as

$$y(t_i) = \sqrt{E_i}x(t_i) + \xi_i \tag{2.3}$$

where $\xi_i$ includes the effects of observation noise and channel distortions. It is assumed that $\xi_i$ is zero mean Gaussian distributed with variance $\sigma^2$.

The sensor operates under the constraint of a fixed power $P_0$. The energy allocated to

the $i$-th sample is thus $E_{0i} = P_0(t_i - t_{i-1})$. Consequently, the transmission energy per sample is $E_i = P_0(t_i - t_{i-1}) - E_c$.

At time $t$, the FC will obtain an estimated level set, $\hat{\mathcal{S}}(t)$, by using the set of discrete-time samples, $\{y(t_i)|t_i \leq t\}$. We define the level set estimation error at time $t$ as the symmetric difference between the level set of interest, $\mathcal{S}(t)$, and the estimated level set, $\hat{\mathcal{S}}(t)$ :

$$e(\hat{\mathcal{S}}(t)) := \int_0^t \mathbb{I}\left\{\mu \in \Delta(\mathcal{S}(t), \hat{\mathcal{S}}(t))\right\} d\mu \tag{2.4}$$

where $\Delta(\mathcal{S}(t), \hat{\mathcal{S}}(t)) = (\mathcal{S}(t) \cap \hat{\mathcal{S}}^c(t)) \cup (\mathcal{S}^c(t) \cap \hat{\mathcal{S}}(t))$ denotes the symmetric difference, $\mathcal{S}^c$ is the complement of $\mathcal{S}$, and $\mathbb{I}\{E\} = 1$ if event $E$ is true and 0 otherwise.

The level set estimation (LSE) problem can then be formulated as

$$\begin{aligned} \min . \quad & \lim_{t\to\infty} \frac{1}{t}\mathbb{E}[e(\hat{\mathcal{S}}(t))] \\ \text{s.t.} \quad & E_i = P_0(t_i - t_{i-1}) - E_c \\ & E_i \geq 0 \end{aligned} \tag{2.5}$$

The optimization problem involves two steps: first, how to choose the sampling instants $\{t_i\}_i$; second, once $\{t_i, y(t_i)\}_i$ is given, how to estimate the level set by using the knowledge of $\{y(t_i)\}_i$.

The choice of the sampling intervals plays a critical role on the LSE performance. Under a fixed power constraint, a larger interval between two consecutive samples means more energy per sample, thus a higher signal-to-noise ratio (SNR) per sample. On the other hand, a larger interval results in weaker temporal correlation between the two samples, and

this might negatively impact the estimation accuracy. Therefore it is important to identify the sequence of sampling intervals that can balance the tradeoff between SNR and sample correlation.

In this paper, we will consider two different sampling scenarios by adding additional constraints to the optimization problem in (2.5).

- **Dynamic active sampling:** After collecting the first $n$ samples, the sensor dynamically selects $t_{n+1}$ based on the knowledge of all previous samples $\{y(t_i)\}_{i=1}^n$ in a myopic manner, such that the expected error up to time $t_{n+1}$, $\frac{1}{t_{n+1}}\mathbb{E}[e(\hat{\mathcal{S}}(t_{n+1}))]$, is minimized.

- **Uniform sampling:** Uniform sampling adds an additional constraint, $t_{i+1} - t_i = d$, $\forall i$, to the optimization problem in (2.5). In this case, we can optimize the value $d$ to minimize the expected estimation error.

Uniform sampling employs a constant sampling period to minimize the global cost function as $t \to \infty$, while the dynamic active sampling adjusts the sampling instant in a myopic manner to minimize the cost function up to the next sampling instant, by using knowledge learned from the previous samples. We will study the design and performance of these two sampling schemes in Sections 2.5 and 3.5, respectively.

Before moving to the two different sampling schemes, we first study in Section 5.5 the optimum estimation of level set once $\{y(t_i)\}_i$ are known at the receiver. The results will provide the analytical form of the conditional LSE error given $\{y(t_i)\}$, which can be used to facilitate the optimum sampling designs in Sections 2.5 and 3.5.

## 2.4 Optimal Level Set Estimation in GP

In this section, we study the optimal level set estimation if the sampling instants, $\mathbf{t}_n = [t_1, \cdots, t_n]$, and the discrete-time samples, $\mathbf{y}_n = [y(t_1), y(t_2), \ldots, y(t_n)]^T$, are known at the FC. This is the operation performed at the FC after the sampling instants, $\{t_n\}_n$, have already been chosen through one of the sampling schemes to be discussed in the next two sections.

Define $\mathbf{r}_{x\mathbf{y}_n}(t) := \mathbb{E}[x(t)\mathbf{y}_n^T] \in \mathcal{R}^n$ and $\mathbf{R}_{\mathbf{y}_n\mathbf{y}_n} := \mathbb{E}[\mathbf{y}_n\mathbf{y}_n^T] \in \mathcal{R}^{n\times n}$, where $\mathcal{R}$ is the set of real numbers. From (2.1) and (2.3), the $i$-th element of the vector $\mathbf{r}_{x\mathbf{y}_n}(t)$ is $\sqrt{E_i}k(t,t_i)$, and the $(i,j)$-th element of the matrix $\mathbf{R}_{\mathbf{y}_n\mathbf{y}_n}$ is $\sqrt{E_iE_j}k(t_i,t_j) + \sigma^2\delta_{ij}$, with $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Due to the GP modelling, given $\mathbf{y}_n$, the distribution of $x(t)$ is still Gaussian, with mean $\hat{m}_n(t)$ and variance $\hat{k}_n(t)$ given by

$$\hat{m}_n(t) = \mathbf{r}_{x\mathbf{y}_n}(t)\mathbf{R}^{-1}_{\mathbf{y}_n\mathbf{y}_n}\mathbf{y}_n \tag{2.6}$$

$$\hat{k}_n(t) = k(t,t) - \mathbf{r}_{x\mathbf{y}_n}(t)\mathbf{R}^{-1}_{\mathbf{y}_n\mathbf{y}_n}\mathbf{r}_{x\mathbf{y}_n}(t)^T. \tag{2.7}$$

The GP regression based LSE algorithm is given in Algorithm 5.

*Theorem 2.1*: Algorithm 1 is optimal with given $\{\mathbf{t}_n, \mathbf{y}_n\}$, i.e., it minimizes the conditional LSE error, $\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_n]$.

**Proof:** Under the GP modelling on $x(t)$ and $y(t)$, $\mathcal{S}(t_n)$ is also a random process. Given the observation history $\mathbf{y}_n$, we can always obtain a posterior distribution of $\mathcal{S}(t_n)$.

**Algorithm 1** GP regression based level set estimation

1: Input: $\mathbf{t}_n$ and $\mathbf{y}_n$
2: Run GP regression for $\forall t \in [0, t_n]$:

$$\hat{m}_n(t) := \mathbf{r}_{x\mathbf{y}_n}(t)\mathbf{R}^{-1}_{\mathbf{y}_n\mathbf{y}_n}\mathbf{y}_n$$

3: Threshold $\hat{m}_n(t)$:

$$\hat{\mathcal{S}}(t_n) = \{t \in [0, t_n] : \hat{m}(t) > \gamma\}$$

4: Output $\hat{\mathcal{S}}(t_n)$.

Given $\mathbf{y}_n$, the expected LSE error can be calculated as

$$\begin{aligned}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_n] &:= \int_0^{t_n} \mathbb{P}\left[t \in \Delta(\mathcal{S}(t_n), \hat{\mathcal{S}}(t_n))\middle|\, \mathbf{y}_n\right] dt \\ &= \int_0^{t_n} \Big(\mathbb{P}\left[x(t) \le \gamma|\, \mathbf{y}_n\right] \cdot \mathbb{I}\{t \in \hat{\mathcal{S}}(t_n)\} \\ &\qquad + \mathbb{P}\left[x(t) > \gamma|\, \mathbf{y}_n\right] \cdot \mathbb{I}\{t \in \hat{\mathcal{S}}^c(t_n)\}\Big) dt \end{aligned} \tag{2.8}$$

Therefore, the optimal estimator that minimizes (4.10) is to let

$$t \in \begin{cases} \hat{\mathcal{S}}(t_n) & \text{if } \mathbb{P}\left[x(t) > \gamma|\mathbf{y}_n\right] > \mathbb{P}\left[x(t) \le \gamma|\mathbf{y}_n\right] \\ \hat{\mathcal{S}}^c(t_n) & \text{if } \mathbb{P}\left[x(t) > \gamma|\mathbf{y}_n\right] \le \mathbb{P}\left[x(t) \le \gamma|\mathbf{y}_n\right] \end{cases} \tag{2.9}$$

for every $t \in [0, t_n]$.

Since $x(t)$ given $\mathbf{y}_n$ is still Gaussian distributed with mean and variance given in (4.8) and (4.9), we have

$$\mathbb{P}\left[x(t) > \gamma|\mathbf{y}_n\right] = Q\left(\frac{\gamma - \hat{m}(t)}{\sqrt{\hat{k}(t)}}\right) \tag{2.10}$$

where $Q(\cdot)$ is the Gaussian-$Q$ function. The optimal estimator defined in (4.11) is then reduced to compare $\hat{m}(t)$ with $\gamma$. If $\hat{m}(t) > \gamma$, the probability in (4.12) is greater than 1/2, thus, we should let $t \in \hat{\mathcal{S}}(t_n)$; otherwise, we let $t \in \hat{\mathcal{S}}^c(t_n)$. ■

A byproduct of the proof of Theorem 3.1 is the conditional LSE error given $\mathbf{y}_n$, and the result is given as follows.

*Corollary 2.1*: The minimum conditional LSE error with given $\mathbf{y}_n$ is

$$\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_n] = \int_0^{t_n} Q\left(\frac{|\gamma - \hat{m}_n(t)|}{\sqrt{\hat{k}_n(t)}}\right) dt \tag{2.11}$$

The results in Theorem 3.1 and Corollary 2.1 will be used to facilitate the designs of dynamic active sampling and uniform sampling in the next two sections.

## 2.5 Dynamic Active Sampling

In this section, we consider the dynamic active sampling scheme, where the FC dynamically selects the next sensing instant $t_n$ based on the sensing history $\mathbf{y}_{n-1}$. With the information extracted from sensing history $\mathbf{y}_{n-1}$, the FC obtains updated information regarding the current function value $x(t)$, which is different from its prior distribution. With such information, the FC can adaptively select $t_n$ in a myopic manner, with the goal to minimize $\frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}]$, the expected normalized level set estimation error up to $t_n$.

The optimization problem in (2.5) in the dynamic active sampling case can be reformulated as

$$\begin{aligned} \text{minimize}_{t_n} \quad & \frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}, t_n] \\ \text{s.t.} \quad & t_n \geq t_{n-1} + d_c \end{aligned} \tag{2.12}$$

where $d_c := \frac{E_c}{P_0}$ is the hardware energy normalized by the power constraint. The cost function, $\frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}, t_n]$, depends on both past observations, $\mathbf{y}_{n-1}$, and potential future sampling instant, $t_n$. It should be noted that value of $y(t_n)$ at $t_n$ is not available during the sensing set selection stage, even if $t_n$ is given.

The cost function in (4.4) can be alternatively expressed as

$$\frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}, t_n] = \frac{1}{t_n}\mathbb{E}_{y_n}\left\{\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_n]\right\} \tag{2.13}$$

The reason that we take another layer of expectation with respect to $y_n$ on the right hand side (RHS) of (4.5) is due to the fact that $y_n$ is unknown before the selection of $t_n$. The value of $y_n$ will be revealed only after sensing sample is collected at time $t_n$.

The observation history up to time $t_{n-1}$ is used to obtain an initial estimate of the distribution of $x(t)$, i.e., its posteriori mean $\bar{m}(t) = \mathbb{E}\left[x(t)|\mathbf{y}_{n-1}\right]$ and covariance $\bar{k}(t,t') = \mathbb{E}\left[x(t)x(t')|\mathbf{y}_{n-1}\right] - \bar{m}(t)\bar{m}(t')$, which can be expressed as

$$\bar{m}(t) == \mathbf{r}_{x\mathbf{y}_{n-1}}\mathbf{R}^{-1}_{\mathbf{y}_{n-1}\mathbf{y}_{n-1}}\mathbf{y}_{n-1} \tag{2.14}$$

$$\bar{k}(t,t') = k(t,t') - \mathbf{r}_{x\mathbf{y}_{n-1}}\mathbf{R}^{-1}_{\mathbf{y}_{n-1}\mathbf{y}_{n-1}}\mathbf{r}^{T}_{x'\mathbf{y}_{n-1}} \tag{2.15}$$

It should be noted that $\hat{m}(t)$ and $\hat{k}(t,t')$ defined in (4.8) and (4.9) are conditioned upon $\mathbf{y}_n$, and they are different from $\bar{m}(t)$ and $\bar{k}(t,t')$ defined in (4.14) and (4.15), which are conditioned upon $\mathbf{y}_{n-1}$. The results in (4.14) and (4.15) provide a rough sketch of the level set. Once $t_n$ is chosen, they can be used in combination with $y(t_n)$ to refine the level set estimation based on Algorithm 5.

we first decompose the estimation error in (4.13) as a function of $(t_n, y_n)$ and $(\mathbf{t}_{n-1}, \mathbf{y}_{n-1})$.

The difference between $(\bar{m}(t), \bar{k}(t,t'))$ and $(\hat{m}(t), \hat{k}(t,t'))$ is directly related to the impacts of selecting $t_n$ on the expected level set estimation error. To identify the relationship, define the following variables.

$$h(t,t_n) := \bar{r}_{xy_n}\bar{r}^{-1}_{y_ny_n}[y_n - \bar{m}(t_n)] \tag{2.16}$$

$$\sigma_h^2(t,t_n) := \bar{r}^2_{xy_n}\bar{r}^{-1}_{y_ny_n} \tag{2.17}$$

where $\bar{r}_{xy_n} := \mathbb{E}(x(t)y(t_n)|\mathbf{y}_{n-1}) = \sqrt{E_n}\bar{k}(t,t_n)$ , and $\bar{r}_{y_ny_n} := \mathbb{E}(y(t_n)y(t_n)|\mathbf{y}_{n-1}) = E_n\bar{k}(t_n,t_n)+\sigma_0$.

With the notation in (4.5) and (2.17), we have the following theorem regarding the decomposition of $\hat{m}(t)$ and $\hat{k}(t,t')$.

*Theorem 2.2*: The posteriori mean and variance, $\hat{m}(t)$ and $\hat{k}(t,t)$, defined in (4.8) and (4.9) can be decomposed in the following form

$$\hat{m}(t) = \bar{m}(t) + h(t,t_n) \tag{2.18}$$

$$\hat{k}(t,t) = \bar{k}(t,t) - \sigma_h^2(t,t_n) \tag{2.19}$$

We use the notation of $\hat{m}(t)$ and $\hat{k}(t,t)$ instead of $\hat{m}_n(t)$ and $\hat{k}_n(t,t)$ in Theorem 4.2 and following part in this section for concise notations.

**Proof:** The proof relies on the conditional distribution of jointly Gaussian distributed random variables. Consider three jointly Gaussian distributed random vectors, $\mathbf{x}$, $\mathbf{z}_1$, and $\mathbf{z}_2$, and we have the following relationship

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1} + \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}_2|\mathbf{z}_1}\boldsymbol{\Sigma}^{-1}_{\mathbf{z}_2\mathbf{z}_2|\mathbf{z}_1}\left(\mathbf{z}_2 - \boldsymbol{\mu}_{\mathbf{z}_2|\mathbf{z}_1}\right), \tag{2.20}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1} - \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}_2|\mathbf{z}_1}\boldsymbol{\Sigma}^{-1}_{\mathbf{z}_2\mathbf{z}_2|\mathbf{z}_1}\boldsymbol{\Sigma}_{\mathbf{z}_2\mathbf{x}|\mathbf{z}_1}, \tag{2.21}$$

where the notations, $\boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}} = \mathbb{E}(\mathbf{a}|\mathbf{b})$ and $\Sigma_{\mathbf{a}\mathbf{b}|\mathbf{c}} = \mathbb{E}[(\mathbf{a} - \boldsymbol{\mu}_{\mathbf{a}|\mathbf{c}})(\mathbf{b} - \boldsymbol{\mu}_{\mathbf{b}|\mathbf{c}})^T|\mathbf{c}]$, are used in the above expressions, with $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ being three random vectors.

If we let $\mathbf{x} = x(t)$, $\mathbf{z}_1 = \mathbf{y}_{n-1}$, and $\mathbf{z}_2 = y_n = x(t_n) + \xi$, then

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \hat{m}(t), \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \hat{k}(t,t) \tag{2.22}$$

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1} = \bar{m}(t), \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1} = \bar{k}(t,t). \tag{2.23}$$

In addition, it can be easily shown that $\mu_{\mathbf{z}_2|\mathbf{z}_1} = \bar{m}(t_n)$, and

$$\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}_2|\mathbf{z}_1} = \bar{r}_{xy_n} \tag{2.24}$$

$$\boldsymbol{\Sigma}_{\mathbf{z}_2\mathbf{z}_2|\mathbf{z}_1} = \bar{r}_{y_n y_n} \tag{2.25}$$

Substituting the above equations into (4.22) and (4.23) yields (4.20) and (4.21). ■

We note that $\bar{m}(t)$ and $\bar{k}(t,t)$ depend on $(\mathbf{t}_{n-1}, \mathbf{y}_{n-1})$ only, $\sigma_h^2(t,t_n)$ depends on $\mathbf{t}_{n-1}$

and $t_n$, while $h(t, t_n)$ depends on $(\mathbf{t}_{n-1}, \mathbf{y}_{n-1})$ and $(t_n, y_n)$. Moreover, given $(\mathbf{t}_{n-1}, \mathbf{y}_{n-1})$ and $t_n$, $h(t, t_n)$ is a Gaussian random variable with zero mean and variance $\sigma_h^2(t, t_n)$, with the randomness contributed by $y(t_n)$. As a result, the results in Theorem 4.2 decompose $\hat{m}(t)$ and $\hat{k}(t, t)$ into two parts, one part depends on $\mathbf{t}_{n-1}$, and the other part depends on $t_n$.

In order to simplify notations, we define

$$\lambda(t) := \frac{\bar{m}(t) - \gamma}{\sqrt{\bar{k}(t, t)}}, \quad \delta(t) := \frac{\sigma_h(t, t_n)}{\sqrt{\bar{k}(t, t)}} \tag{2.26}$$

With the decomposition given in Theorem 4.2, we have the following results regarding the cost function in (4.4).

*Theorem 2.3*: The cost function in (4.4), $\mathcal{J}(t_n) := \frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}, t_n]$, with respect to the sensing location $t_n$ is

$$\mathcal{J}(t_n) = \frac{1}{\pi t_n} \int_0^{t_n} \int_0^{\frac{\pi}{2}} \left[1 + \frac{\delta^2(t)}{(1 - \delta^2(t)) \sin^2(\theta)}\right]^{-1/2} \times \exp\left\{-\frac{\lambda^2(t)}{2[\delta^2(t) + (1 - \delta^2(t)) \sin^2(\theta)]}\right\} d\theta dt \tag{2.27}$$

where $\lambda(t)$ depends on $(\mathbf{t}_{n-1}, \mathbf{y}_{n-1})$ only and $\delta(t)$ depends on $\mathbf{y}_{n-1}$ and $t_n$ as defined in (4.28).

**Proof:** From (4.5), Corollary 2.1, and Theorem 4.2, the cost function in (4.4) can be alternatively expressed as

$$\frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}, t_n] = \frac{1}{t_n}\mathbb{E}_Y[Q(|Y|)] \tag{2.28}$$

where

$$Y = \frac{h(t,t_n) + \bar{m}(t) - \gamma}{\sqrt{k(t,t) - \sigma_h^2(t,t_n)}}. \tag{2.29}$$

Since $h(t,t_n) \sim \mathcal{N}(0, \sigma_h^2(t,t_n))$, it is straightforward that the random variable $Y$ is Gaussian distributed with mean $\mu_Y = \frac{\lambda(t)}{\sqrt{1-\delta^2(t)}}$ and variance $\sigma_Y^2 = \frac{\delta^2(t)}{1-\delta^2(t)}$.

With Craig's alternative expression of the $Q$-function [3], we have

$$\mathbb{E}_Y[Q(|Y|)] = \frac{1}{\pi}\int_0^{\frac{\pi}{2}} \mathbb{E}_Y\left[e^{-\frac{Y^2}{2\sin^2(\theta)}}\right] d\theta \tag{2.30}$$

Since $Y$ is Gaussian distributed, $Z := Y^2/\sigma_Y^2$ is non-central $\chi^2$-distributed with one degree-of-freedom. The moment generating function (MGF) of $Z$ is

$$M_Z(s) = \mathbb{E}\left[e^{\frac{sY^2}{\sigma^2}}\right] = \exp\left\{\frac{\mu_Y^2 s}{(1-2s)\sigma_Y^2}\right\}\frac{1}{\sqrt{1-2s}} \tag{2.31}$$

Combining (4.32) with (4.33) yields

$$\mathbb{E}_Y[Q(|Y|)] = \frac{1}{\pi}\int_0^{\frac{\pi}{2}} \exp\left\{-\frac{\mu_Y^2}{2(\sigma_Y^2 + \sin^2(\theta))}\right\}\frac{1}{\sqrt{1+\frac{\sigma_Y^2}{\sin^2(\theta)}}} d\theta$$

This completes the proof. ■

Theorem 4.3 gives the exact explicit expression of the cost function of the optimization problem. It is expressed as a double integral, and might be difficult to evaluate. To simplify calculation, we develop an upper bound of the cost function. The upper bound is obtained by applying $\sin\theta \leq 1$ to (4.29), and the result is given in Corollary 2.2.

*Corollary 2.2*: The cost function in (4.4) with respect to the sensing time $t_n$ is upper bounded by

$$\frac{1}{t_n}\mathbb{E}[e(\hat{\mathcal{S}}(t_n))|\mathbf{y}_{n-1}, t_n] \leq \frac{1}{2t_n}\int_0^{t_n} \exp\left\{-\frac{\lambda^2(t)}{2}\right\} \cdot \tau(t)dt \tag{2.32}$$

where $\tau(t) = \sqrt{1-\delta^2(t)} = \sqrt{1-\frac{\sigma_h^2(t,t_n)}{\bar{k}(t,t)}}$.

In the upperbound (4.34), the variables $\lambda(t)$ and $\bar{k}(t,t)$ are independent of the choice of $t_n$, and the choice of $t_n$ will only impact $\sigma_h^2(t,t_n)$.

We propose to minimize the upper bound given in (4.34). Define $\alpha(t) = \frac{1}{2}\exp\left\{-\frac{\lambda^2(t)}{2}\right\}$. Then, a new optimization problem with the upper bound being the cost function can be formulated as

$$\begin{aligned} \min_{t_n} \quad & \bar{\mathcal{J}}(t_n) := \frac{1}{t_n}\int_0^{t_n} \alpha(t)\sqrt{1-\frac{\sigma_h^2(t,t_n)}{\bar{k}(t,t)}}dt \\ \text{s.t.} \quad & t_n \geq t_{n-1} + d_c \end{aligned} \tag{2.33}$$

In the integrand in the objective function in (4.35), the weight $\alpha(t)$ is a function of $\lambda(t)$, which is defined in (4.28). We note that in its definition, the numerator $|\bar{m}(t) - \gamma|$ measures the deviation of $\bar{m}(t)$ from the threshold $\gamma$, which is then normalized by $\sqrt{\bar{k}(t,t)}$, the estimated standard deviation. The larger the value of $\lambda(t)$, the less likely an classification error will happen at time $t$. This is reflected by $\alpha(t)$, since it is decreasing in $\lambda(t)$. With a small weight $\alpha(t)$, the term $\alpha(t)\tau(t)$ plays a less important role in the optimization (4.35). The solution to (4.35) thus automatically allocates more resources to the locations with larger $\alpha(t)$.

Since the cost function $\bar{\mathcal{J}}(t_n)$ in (4.35) is continuous and differentiable with respect to $t_n$, the optimum value of $t_n$ must be one of the zero-slope points of $\bar{\mathcal{J}}(t_n)$. Therefore, the optimum value must be one of the solutions to $\bar{\mathcal{J}}'(t_n) = 0$,

which can be expressed as

$$\begin{aligned}\alpha(t_n)\sqrt{1-\frac{\sigma_h^2(t_n,t_n)}{\bar{k}(t_n,t_n)}} &= \frac{1}{t_n}\int_0^{t_n}\alpha(t)\sqrt{1-\frac{\sigma_h^2(t,t_n)}{\bar{k}(t,t)}}dt \\ &+\frac{1}{2}\int_0^{t_n}\frac{\alpha(t)}{\sqrt{\bar{k}(t,t)\big[\bar{k}(t,t)-\sigma_h^2(t,t_n)\big]}}\frac{\partial\sigma_h^2(t,t_n)}{\partial t_n}dt \end{aligned} \tag{2.34}$$

where

$$\frac{\partial\sigma_h^2(t,t_n)}{\partial t_n} = \frac{\bar{k}^2(t,t_n)\big[\frac{\sigma^2}{E_n}(2\log\rho+1)+2\log\rho\big]}{\big[\bar{k}(t_n,t_n)+\frac{\sigma^2}{E_n}\big]^2} \tag{2.35}$$

The above problem can be solved numerically with the fsolve function in Matlab. Our numerical results indicate there is always just a unique solution to the above equation for all configurations considered in this paper.

The dynamic active sampling picks the sampling sequentially in a myopic manner. It attempts to minimize the cost function evaluated up to the next sampling instant. Next we will study the uniform sampling scheme, which tries to minimize the global cost function evaluated as $t \to \infty$.

## 2.6 Optimal Uniform Sampling

In this section, optimal uniform sampling scheme will be studied. we will first find the unconditional LSE error probability using the results from Algorithm 5. The analytical

results will then be used to identify the optimum sampling rate that can minimize the LSE error probability.

### 2.6.1 LSE Error Probability in Uniform Sampling

With uniform sampling, the sensor takes samples at uniform intervals with sampling period $d = t_{i+1} - t_i$. The dimension of the problem is reduced significantly, that is, the optimization variable is reduced to a single variable $d$, as against the time sequence $\{t_i\}_i$ in the original problem.

With the additional constraint of uniform sampling, the optimization problem in (2.5) can be reformulated as

$$\begin{aligned} \min . \quad & \lim_{n\to\infty} \frac{1}{nd}\mathbb{E}[e(\hat{\mathcal{S}}(nd))] \\ \text{s.t.} \quad & t_{i+1} - t_i = d \\ & E := E_i = P_0 d - E_c \\ & d \geq \frac{P_0}{E_c} \end{aligned} \tag{2.36}$$

The unconditional LSE error probability in (3.5) can be alternatively expressed as

$$\mathbb{E}[e(\hat{\mathcal{S}}(t))] = \mathbb{E}_{\mathbf{y}_n}\left\{\mathbb{E}[e(\hat{\mathcal{S}}(nd))|\mathbf{y}_n]\right\} \tag{2.37}$$

From (3.12) and Corollary 2.1, the cost function depends on the posteriori mean $\hat{m}_n(t)$

and variance $\hat{k}_n(t)$. From (4.8) and (4.9), we have

$$\hat{m}_n(t) = \sqrt{E}\mathbf{r}_{x\mathbf{x}_n}(t)\left(E\,\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n} + \sigma^2\mathbf{I}_n\right)^{-1}\mathbf{y}_n \tag{2.38}$$

$$\hat{k}_n(t) = k(t,t) - E\,\mathbf{r}_{x\mathbf{x}_n}(t)\left(E\,\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n} + \sigma^2\mathbf{I}_n\right)^{-1}\mathbf{r}^T_{x\mathbf{x}_n}(t). \tag{2.39}$$

where $\mathbf{r}_{x\mathbf{x}_n}(t) = \mathbb{E}\left[x(t)\mathbf{x}_n\right] \in \mathcal{R}^n$ and $\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n} = \mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T] \in \mathcal{R}^{n\times n}$, with $\mathbf{x}_n = [x(d), x(2d), \cdots, x(nd)]^T \in \mathcal{R}^n$.

The posteriori mean is a function of $\mathbf{y}_n$, whereas the posteriori variance is a constant independent of $\mathbf{y}_n$. Since $\mathbf{y}_n$ is zero-mean Gaussian distributed, it can be easily shown that $\hat{m}_n(t)$ is zero-mean Gaussian distributed with variance being

$$\sigma^2_{\hat{m}_n}(t) = E\,\mathbf{r}_{x\mathbf{x}_n}(t)\left(E\,\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n} + \sigma^2\mathbf{I}_n\right)^{-1}\mathbf{r}^T_{x\mathbf{x}_n}(t) \tag{2.40}$$

From (3.14) and (3.15), we have $\hat{k}_n(t) = k(t,t) - \sigma^2_{\hat{m}}(t)$.

The variance in (3.15) depends on a number of factors, such as the correlation coefficient $\rho$, the sampling period $d$, the energy per sample $E$, and the time instant $t$. As $n \to \infty$, we have the following asymptotic results of $\sigma^2_{\hat{m}_n}(t)$ and $\hat{k}_n(t)$.

*Theorem 2.4*: Define the asymptotic posteriori variance $\sigma^2_e(\mu) := \lim_{n\to\infty}\hat{k}_n(t)$, where $\mu = \frac{t}{d} - \lfloor\frac{t}{d}\rfloor \in [0,1]$ is the relative position of $t$ between two adjacent samples. We have

$$\sigma^2_e(\mu) = \left[\frac{1}{\gamma_0(d-d_c)} + \frac{1+\rho^{2d}-\rho^{2\mu d}-\rho^{2(1-\mu)d}}{1-\rho^{2d}}\right] \left(\frac{1}{\gamma_0(d-d_c)} + \frac{1-\rho^d}{1+\rho^d}\right)^{-\frac{1}{2}} \left(\frac{1}{\gamma_0(d-d_c)} + \frac{1+\rho^d}{1-\rho^d}\right)^{-\frac{1}{2}} \tag{2.41}$$

where $\gamma_0 := \frac{P_0}{\sigma_2}$ being the SNR, and $d_c := \frac{E_c}{P_0}$ is hardware energy normalized by the average power constraint. In addition,

$$\lim_{n\to\infty} \sigma^2_{\hat{m}_n}(t) = 1 - \sigma^2_e(\mu). \tag{2.42}$$

**Proof:** Define a new vector, $\mathbf{x}'_n = [x(d+\mu d), x(2d+\mu d), \cdots, x(nd+\mu d)]^T$, which is obtained by shifting $\mathbf{x}_n$ to the right by $\mu d$ seconds. The posterior covariance matrix of $\mathbf{x}'_n$ given $\mathbf{y}_n$ is $\hat{\mathbf{R}}_{\mathbf{x}'_n\mathbf{x}'_n} := \mathbb{E}[\mathbf{x}'_n\mathbf{x}'^T_n|\mathbf{y}_n]$, which can be expressed as

$$\hat{\mathbf{R}}_{\mathbf{x}'_n\mathbf{x}'_n} = \mathbf{R}_{\mathbf{x}'_n\mathbf{x}'_n} - E\,\mathbf{R}_{\mathbf{x}'_n\mathbf{x}_n}\left(E\,\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n} + \sigma^2\mathbf{I}_n\right)^{-1}\mathbf{R}^T_{\mathbf{x}'_n\mathbf{x}_n} \tag{2.43}$$

where $\mathbf{R}_{\mathbf{x}'_n\mathbf{x}'_n} = \mathbb{E}[\mathbf{x}'_n\mathbf{x}'^T_n] = \mathbf{R}_{\mathbf{x}_n\mathbf{x}_n}$ and $\mathbf{R}_{\mathbf{x}'_n\mathbf{x}_n} = \mathbb{E}[\mathbf{x}'_n\mathbf{x}^T_n]$. Since the value of $\hat{k}_n(t)$ in (3.14) is on the diagonal of $\hat{\mathbf{R}}_{\mathbf{x}'_n\mathbf{x}'_n}$, we have

$$\sigma^2_e(\mu) = \lim_{n\to\infty} \hat{k}_n(t) = \lim_{n\to\infty} \frac{1}{n}\text{trace}(\hat{\mathbf{R}}_{\mathbf{x}'_n\mathbf{x}'_n}) \tag{2.44}$$

From (2.1), $\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n}$ is a symmetric Toeplitz matrix with the $(i,j)$-th element being $\rho^{|i-j|d}$. Similarly, the matrix $\mathbf{R}_{\mathbf{x}'_n\mathbf{x}_n}$ is an asymmetric Toeplitiz matrix with the first row being $[\rho^{|\mu d|}, \rho^{|\mu-1|d}, \cdots, \rho^{|\mu-(n-1)|d}]$, and the first column $[\rho^{|\mu d|}, \rho^{|\mu+1|d}, \cdots, \rho^{\mu+(n-1)|d}]^T$.

The Toeplitz matrix, $\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n}$ is uniquely determined by the sequence $\{\rho^{|n|d}\}_n$, whose discrete-time Fourier transform (DTFT), $\Psi_{\mathbf{x}_n\mathbf{x}_n}(\omega) = \sum_{n=-\infty}^{+\infty} \rho^{|n|d}e^{-jn\omega}$, is

$$\Psi_{\mathbf{x}_n\mathbf{x}_n}(\omega) = \frac{1-\rho^{2d}}{1+\rho^{2d}-2\rho^d\cos(\omega)} \tag{2.45}$$

The Toeplitz matrix, $\mathbf{R}_{\mathbf{x}'_n\mathbf{x}_n}$, is uniquely determined by the sequence, $\{\rho^{|\mu+n|d}\}_n$. The DTFT, $\Psi_{\mathbf{x}'_n\mathbf{x}_n}(\omega) = \sum_{n=-\infty}^{+\infty} \rho^{|\mu+n|d} e^{-jn\omega}$, can be calculated as

$$\Psi_{\mathbf{x}'_n\mathbf{x}_n}(\omega) = \frac{\rho^{-\mu d}\left[\rho^d(1-\rho^{2\mu d})e^{j\omega} + \rho^{2\mu d} - \rho^{2d}\right]}{1+\rho^{2d}-2\rho^d\cos(\omega)}. \tag{2.46}$$

Based on [5, Lemma 2], $\mathbf{R}_{\mathbf{x}'_n\mathbf{x}_n}$ is asymptotically equivalent to a circulant matrix, $\mathbf{C}_{\mathbf{x}'_n\mathbf{x}_n} = \mathbf{U}_n^H \mathbf{D}_{\mathbf{x}'_n\mathbf{x}_n}\mathbf{U}_n$, where $\mathbf{U}_n$ is the unitary discrete Fourier transform (DFT) matrix with the $(i,l)$-th element being $(\mathbf{U}_n)_{i,l} = \frac{1}{\sqrt{n}}\exp\left[-j2\pi\frac{(i-1)(l-1)}{n}\right]$, and $\mathbf{D}_{\mathbf{x}'_n\mathbf{x}_n}$ is a diagonal matrix with its $k$-th element being $(\mathbf{D}_{\mathbf{x}'_n\mathbf{x}_n})_k = \Psi_{\mathbf{x}'_n\mathbf{x}_n}(2\pi\frac{k-1}{n})$.

Similarly, the Toeplitz matrix, $\mathbf{R}_{\mathbf{x}_n\mathbf{x}_n}$, is asymptotically equivalent to a circulant matrix, $\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n} = \mathbf{U}_n^H \mathbf{D}_{\mathbf{x}_n\mathbf{x}_n}\mathbf{U}_n$, where $\mathbf{D}_{\mathbf{x}_n\mathbf{x}_n}$ is a diagonal matrix with its $k$-th element being $(\mathbf{D}_{\mathbf{x}_n\mathbf{x}_n})_k = \Psi_{\mathbf{x}_n\mathbf{x}_n}(2\pi\frac{k-1}{n})$.

Based on [8, Theorem 2.1], $\hat{\mathbf{R}}_{\mathbf{x}'_n\mathbf{x}'_n}$ in (2.43) is asymptotically equivalent to a circulant matrix $\hat{\mathbf{C}}_{\mathbf{x}'_n\mathbf{x}'_n} = \mathbf{C}_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{C}_{\mathbf{x}'_n\mathbf{x}_n}(\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n} + \frac{1}{\gamma_0(d-d_c)})^{-1}\mathbf{C}^H_{\mathbf{x}'_n\mathbf{x}_n} = \mathbf{U}_n^H\hat{\mathbf{D}}_{\mathbf{x}_n\mathbf{x}_n}\mathbf{U}_n$, where $\hat{\mathbf{D}}_{\mathbf{x}_n\mathbf{x}_n} = \mathbf{D}_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{D}_{\mathbf{x}'_n\mathbf{x}_n}(\mathbf{D}_{\mathbf{x}_n\mathbf{x}_n} + \frac{1}{\gamma_0(d-d_c)})^{-1}\mathbf{D}^H_{\mathbf{x}'_n\mathbf{x}_n}$.

From Szego's Theorem [5], we have

$$\lim_{n\to\infty}\hat{k}_n(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\Psi_{\mathbf{x}_n\mathbf{x}_n}(\omega) - \frac{|\Psi_{\mathbf{x}'_n\mathbf{x}_n}(\omega)|^2}{\Psi_{\mathbf{x}_n\mathbf{x}_n}(\omega)+\frac{1}{\gamma_0(d-d_c)}}\right]d\omega. \tag{2.47}$$

Substituting (2.45) and (2.46) into above equation and solving the integral with [21, eqn. (2553.3)], we have the result in (2.41). ■

In Theorem 3.2, the asymptotic posteriori variance $\sigma_e^2(\mu)$ is expressed as an explicit function of a number of parameters, such as the SNR $\gamma_0$, the sampling period $d$, the normalized

hardware energy $d_c$, the temporal correlation coefficient $\rho$, and the relative time location $\mu$.

From Theorem 3.2, as $n \to \infty$, $\hat{m}_n(kd+\mu)$ is a zero-mean Gaussian random variable with variance $1-\sigma_e^2(\mu)$. Therefore, the statistical properties of $\hat{m}_n(t)$ as $n \to \infty$ are periodic in $t$ with period $d$. Define $\hat{m}(\mu) := \lim_{n\to\infty} \hat{m}_n(kd+\mu)$. We have the following corollary regarding the distribution of $\hat{m}(\mu)$.

*Corollary 2.3*: As $n \to \infty$, $\hat{m}(\mu) = \lim_{n\to\infty} \hat{m}_n(kd+\mu)$ is zero-mean Gaussian distributed with variance $1-\sigma_e^2(\mu)$, that is, $\hat{m}(\mu) \sim \mathcal{N}(0, 1-\sigma_e^2(\mu))$.

With the asymptotic results in Theorem 3.2 and Corollary 3.1, we can get the explicit expression of the cost function in (3.5), and the result is given in the following theorem.

*Theorem 2.5*: The cost function in (3.5), $\mathcal{J} := \lim_{n\to\infty} \frac{1}{nd}\mathbb{E}[e(\hat{\mathcal{S}}(nd))]$, can be expressed as

$$\mathcal{J} = \frac{1}{\pi}\int_0^1 \int_0^{\pi/2} \left[1 + \frac{1-\sigma_e^2(\mu)}{\sigma_e^2(\mu)\sin^2(\theta)}\right]^{-1/2} \times \exp\left(-\frac{\gamma^2/2}{1-\sigma_e^2(\mu)\cos^2(\theta)}\right) d\mu d\theta. \tag{2.48}$$

**Proof:** We first consider the conditional cost function, $\lim_{n\to\infty} \frac{1}{nd}\mathbb{E}[e(\hat{\mathcal{S}}(nd))|\mathbf{y}_n]$. From (4.13) and (3.12), the cost function can be expressed as

$$\lim_{n\to\infty} \frac{1}{nd}\mathbb{E}[e(\hat{\mathcal{S}}(nd))|\mathbf{y}_n] = \lim_{n\to\infty} \frac{1}{nd}\sum_{i=1}^{n} \int_{(i-1)d}^{id} Q\left(\frac{|\gamma - \hat{m}_n(t)|}{\sqrt{\hat{k}_n(t)}}\right) dt$$

Performing change of variable, $t = (i-1)d + \mu d$, in the above integral yields, and using the

results from Theorem 3.2, we have

$$\lim_{n\to\infty}\frac{1}{nd}\mathbb{E}[e(\hat{\mathcal{S}}(nd))|\mathbf{y}_n] = \int_0^1 Q\left(\frac{|\gamma-\hat{m}(\mu)|}{\sqrt{\sigma_e^2(\mu)}}\right)d\mu \tag{2.49}$$

In the conditional cost function in (3.27), there is only one random variable, $\hat{m}(\mu) \in \mathcal{N}(0, 1-\sigma_e^2(\mu))$, which is a function of $\mathbf{y}_n$. Therefore the unconditional cost function can be expressed as

$$\mathcal{J}=\lim_{n\to\infty}\frac{1}{nd}\mathbb{E}[e(\hat{\mathcal{S}}(nd))]=\int_0^1 \mathbb{E}_{\hat{m}(\mu)}\left[Q\left(\frac{|\gamma-\hat{m}(\mu)|}{\sqrt{\sigma_e^2(\mu)}}\right)\right]d\mu \tag{2.50}$$

With Craig's alternative expression of the $Q$-function [3], (3.28) can be reformulated as

$$\mathcal{J} = \frac{1}{\pi}\int_0^1\int_0^{\frac{\pi}{2}} \mathbb{E}_{\hat{m}(\mu)}\left[e^{-\frac{(\hat{m}(\mu)-\gamma)^2}{2\sigma_e^2(\mu)\sin^2(\theta)}}\right]d\mu d\theta \tag{2.51}$$

Define $Z := \frac{(\hat{m}(\mu)-\gamma)^2}{1-\sigma_e^2(\mu)}$, which is a non-central $\chi^2$-distributed random variable with one degree-of-freedom and the non-centrality parameter $\frac{\gamma^2}{1-\sigma_e^2(\mu)}$. The MGF of $Z$ is given in (4.33). Combining (3.29) with (4.33) yields (3.26). ■

The results in Theorem 3.3 give the exact analytical expression of the cost function, which is expressed as a function of the optimization parameter $d$, and other system parameters such as the SNR $\gamma_0$, the temporal correlation coefficient $\rho$, and the normalized hardware energy consumption $d_c$. Thus, given $\{\gamma_0, \rho, d_c\}$, we can identify $d \geq d_c$ that minimizes the cost function $\mathcal{J}$ in (3.26). The integrand in (3.26) has only elementary functions, and the integration limits are finite. Thus the integrals in (3.26) can be easily evaluated with numerical integrations with high precision.

### 2.6.2 Optimum Sampling Rate

Due to the complicated form of the double integrals in (3.26), it might be difficult to directly minimize the exact cost function with respect to $d$. We resort to an upper bound of $d$ to simplify the optimization.

The following corollary provides an upper bound of the cost function expressed in a closed-form.

*Corollary 2.4*: The asymptotical expected LSE error in (3.26) is upper bounded by

$$g(d) = \frac{1}{2}\exp\left(-\frac{\gamma^2}{2}\right)\left[\frac{1}{\gamma_0(d-d_c)} + \frac{1+\rho^{2d}}{1-\rho^{2d}} + \frac{1}{d\log\rho}\right]^{1/2} \left(\frac{1}{\gamma_0(d-d_c)} + \frac{1-\rho^d}{1+\rho^d}\right)^{-\frac{1}{4}} \left(\frac{1}{\gamma_0(d-d_c)} + \frac{1+\rho^d}{1-\rho^d}\right)^{-\frac{1}{4}} \tag{2.52}$$

**Proof:** It can be easily shown that the integrand in (3.26) is an increasing function with respect to $\theta$ for $\theta \in [0, \pi/2]$. Thus we have

$$\mathbb{E}[e(\hat{\mathcal{S}})] \leq \frac{1}{2}\exp\left(-\frac{\gamma^2}{2}\right)\int_0^1 \sqrt{\sigma_e^2(\mu)}d\mu. \tag{2.53}$$

Due to the concavity of $\sqrt{x}$, the asymptotic LSE error can be upper-bounded again as

$$\mathbb{E}[e(\hat{\mathcal{S}})] \leq \frac{1}{2}\exp\left(-\frac{\gamma^2}{2}\right)\sqrt{\int_0^1 \sigma_e^2(\mu)d\mu}. \tag{2.54}$$

Substituting (2.41) into (2.54) and simplifying yields (2.52). ■

Given the complicated form of the exact LSE error probability in (3.26), we propose to instead minimize the error probability upper bound in (2.52). It will be shown in the

numerical results that minimizing the exact error probability or its upper bound yields almost the same values of the optimum sampling rates.

Since the upper bound in (2.53) is continuous and differentiable with respect to $d$, the optimum value of $d$ that minimizes $g(d)$ must be one of the zero-slope points of $g(d)$. Therefore, the optimum value of $d$ must be one of the solutions to $g'(d) = 0$, which can be expressed as $g'(d) = c(d) \cdot w(d)$, where

$$\begin{aligned} w(d) = & -8 \ln \rho^{-1} \rho^{2d} (1-\rho^{2d})^{-3} + \Bigg[ -4 \ln \rho^{-1} \gamma_0 (d-d_c) \rho^{2d} \\ & + 4 \ln \rho^{-1} \rho^{2d} + \frac{4}{d} \rho^{2d} + \frac{4}{d-d_c} \Bigg] (1-\rho^{2d})^{-2} \\ & + \left[ -\frac{2}{\ln \rho^{-1} d(d-d_c)} - \frac{4}{d-d_c} + \frac{4}{\ln \rho^{-1} d^2} \right] (1-\rho^{2d})^{-1} \\ & + \frac{1}{\ln \rho^{-1} d(d-d_c)} \left( 1 - \frac{1}{\gamma_0 (d-d_c)} \right) \\ & + \frac{1}{d^2 \ln \rho^{-1}} \left( \frac{1}{\gamma_0 (d-d_c)} + \gamma_0 (d-d_c) - 2 \right) \end{aligned} \tag{2.55}$$

and

$$\begin{aligned} c(d) = & \frac{1}{8} \exp(-\gamma^2) g^{-1}(d) \frac{1}{\gamma_0 (d-d_c)} \Bigg[ \Big( \frac{1}{\gamma_0 (d-d_c)} - 1 \Big)^2 \\ & + \frac{4}{\gamma_0 (d-d_c)(1-\rho^{2d})} \Bigg]^{-\frac{3}{2}} > 0 \end{aligned}$$

for all $d > d_c$.

Again the above problem can be solved numerically with the fsolve function in Matlab. Our numerical results indicate the LSE error probability upper bound, $g(d)$, defined in (2.53) is quasi-convex in $d$ and there is always just a unique solution to $w(d) = 0$ for all

configurations considered in this paper.

## 2.7 Numerical and Simulation Results

In this section, numerical and simulation results are presented to demonstrate the performance of the proposed level set sensing and estimation algorithms in both uniform sampling and active sampling scenarios. Without loss of generality, the level set threshold $\gamma$ is set to 0.1.

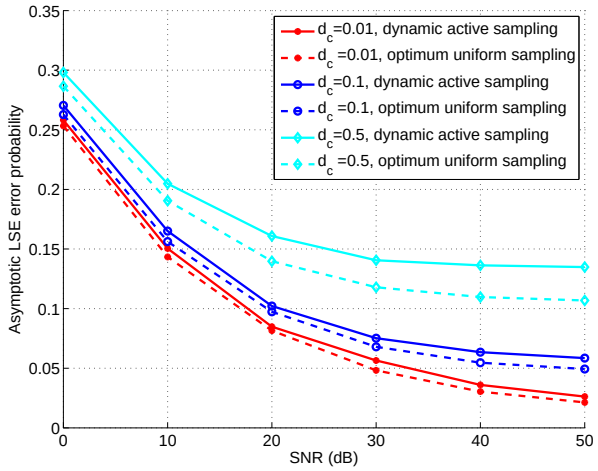



Figure 2.1: LSE error probability as a function of SNR under various $d_c$ ($\rho = 0.5$).

Figs. 2.1 and 2.2 compare the LSE error probabilities between dynamic active sampling and uniform sampling. For the dynamic active sampling, the optimum value of $t_n$ is obtained by solving (2.34); for the uniform sampling, the optimum value of $d$ is calculated by equaling (2.55) zero. Each point in the simulation results are obtained by averaging over 500 independent trials. The LSE error probability are calculated over a time period of $t = 150$ seconds, and both algorithms converge in this time frame. In Fig. 2.1, the power law coefficient is $\rho = 0.5$. In Fig. 2.2, the SNR is $\gamma_0 = 20$ dB. The performance of uniform sampling consistently outperforms that of dynamic active sampling for all system configurations. The

performance difference between the two narrows as the normalized hardware energy consumption $d_c$ decreases. The performance of both schemes degrades as $d_c$ increases, because more energy is consumed by the hardware.

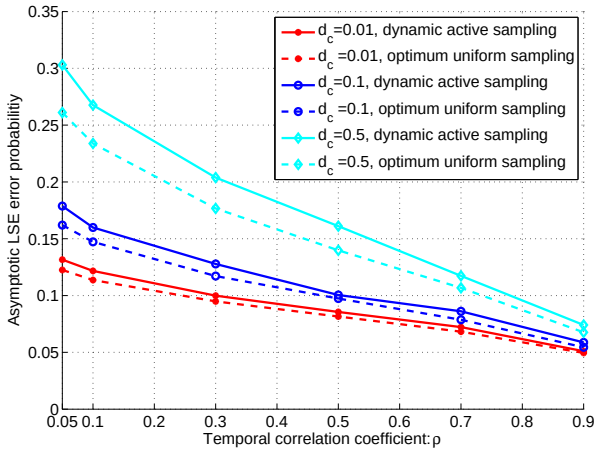



Figure 2.2: LSE error probability as a function of $\rho$ under various $d_c$ ($\gamma_0 = 20$ dB).

Fig. 2.3 compares sampling rates between dynamic active sampling and uniform sampling. For the dynamic active sampling, the value of sampling rates is obtained by dividing the accumulated number of samples in the time frame by the duration of this time frame. For the uniform sampling, the optimum value of sampling rate is the reciprocal of the optimum $d$ . In terms of sampling rate, there is always a gap between the two sampling schemes for all system configurations. The optimum uniform sampling yields more sampling actions than the dynamic active sampling in the same time frame, which partly explains its better performance in term of LSE error probability shown in Figs. 2.1 and 2.2.

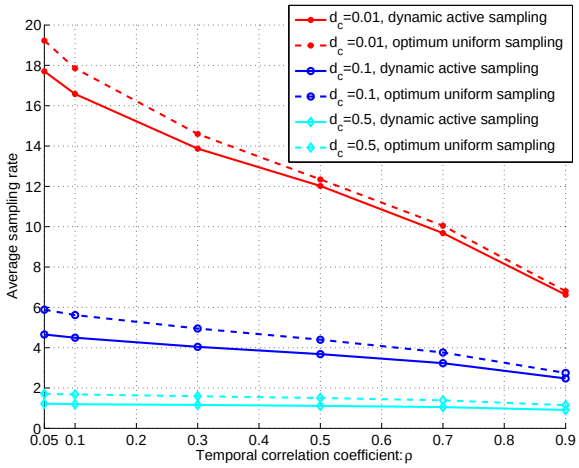



Figure 2.3: Sampling rate as a function of $\rho$ under various $d_c$ ($\gamma_0 = 20$ dB).

Fig. 2.4 shows the asymptotic LSE error probabilities and their corresponding upper bounds of uniform sampling as a function of the sampling rate $r = \frac{1}{d} \leq \frac{1}{d_c}$. The SNR is $\gamma_0 = 10$ dB, and the normalized hardware energy consumption $d_c$ is fixed as 0.05, which corresponds to a maximum sampling rate of $r_{\max} = 20$ Hz. The simulation results are obtained with $n = 100$ samples yet the analytical results are derived by using $n \to \infty$. The simulation results with finite $n$ match very well with the analytical results with infinite $n$, thus the asymptotic analytical results provide a very good approximation of the performance of practical systems with a finite $n$. For all system configurations, when the sampling rate approaches its boundaries at 0 or $r_{\max} = 20$ Hz, the LSE error probability approaches $Q(|\gamma|)$, the error probability of random decisions. At 0 Hz, no sample is collected by the sensor. At $r_{\max}$, all energy is consumed by the sensing operation thus no information is transmitted to the FC. The optimum sampling rates that minimize the error probability upper bound in (2.52) are obtained by equaling (2.55) zero and marked in the figure. The optimum sampling rates that minimize the exact error probability in (3.26) are obtained through exhaustive search. It is observed that the sampling rates that minimize the upper bound or the exact

expression are almost the same. For example, when $\rho = 0.9$, the two optimum sampling rates are 1.85 Hz and 1.81 Hz, respectively. Therefore minimizing the upper bound provides a reasonably accurate approximation of the true optimum sampling rate.

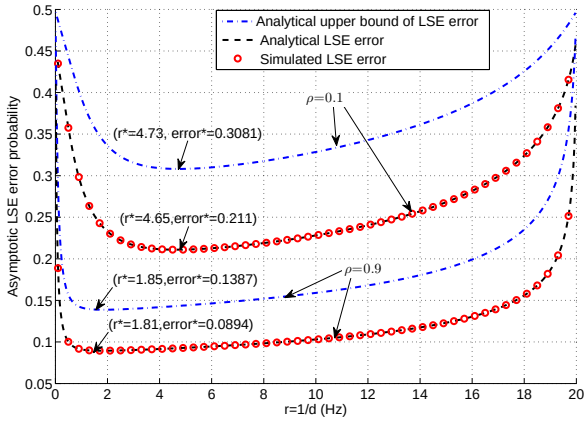



Figure 2.4: LSE error probability as a function of sampling rate $r = 1/d$ for systems with uniform sampling ($\gamma_0 = 10$ dB, $d_c = 0.05$ seconds).

Fig. 2.5 shows the optimum sampling periods $d^*$ as a function of $d_c$ for uniform sampling. The SNR is set as $\gamma_0 = 10$ dB. The results that minimize the upper bound in (2.52) or the exact error probability in (3.26) are shown in the figure. Again, minimizing the upper bound or the exact error probability yields almost identical optimum sampling periods, for all system configurations. The optimum sampling period is an increasing function in $\rho$, in that a larger $\rho$ renders a stronger correlation between two adjacent samples. It is also an increasing function in $d_c$, because more energy needs to be allocated for each sample with a higher hardware energy consumption.

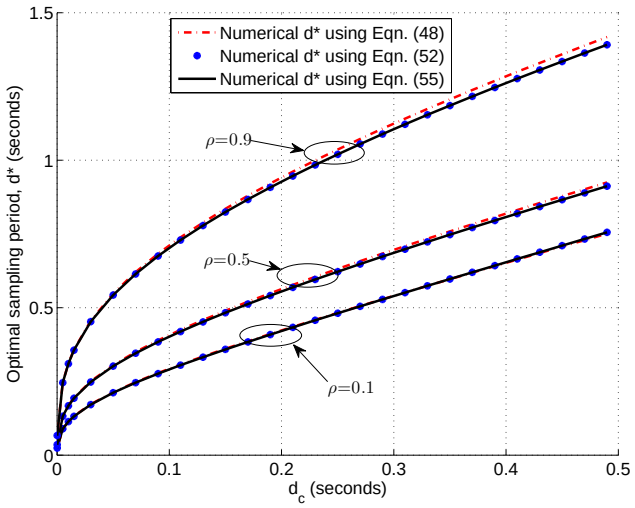



Figure 2.5: Optimum sampling period $d^*$ as a function of $d_c$ for systems with uniform sampling ($\gamma_0 = 10$ dB).

The minimum LSE error probabilities and their upper bounds are shown in Fig. 2.6 as a function of the SNR for systems with uniform sampling. The results are obtained by first identifying the optimum sampling period $d^*$ by zeroing (2.55), and then plugging the values in (3.26) or (2.52). The value of $\rho$ is 0.8. As expected, a higher $d_c$ or a higher hardware energy consumption always results in a higher LSE error probability. The impact of $d_c$ on the LSE is very small when the SNR is low, and it is more pronounced at high SNR. In addition, the gap between the minimum LSE error probability and its upper bound narrows as SNR increases.

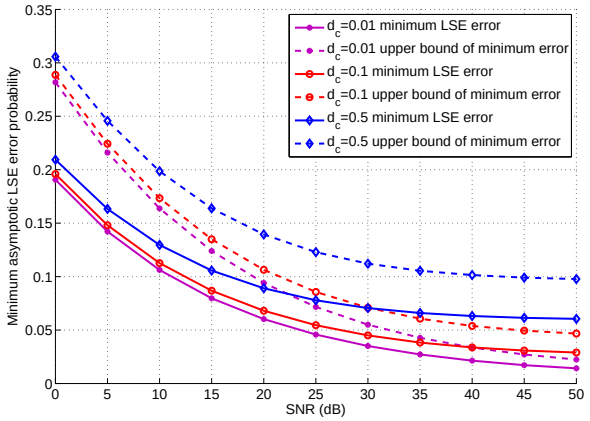



Figure 2.6: Minimum LSE error probabilities as a function of SNR ($\gamma_0$) for systems with uniform sampling ($\rho = 0.8$).

## 2.8 Conclusions

The optimum level set estimations of a time-varying random field with a wireless sensor under a power constraint has been studied in this paper. The optimum designs were performed to minimize the time-averaged LSE error probability by choosing a sequence of sampling instants. Two sampling schemes have been considered, a dynamic active sampling scheme that adaptively selects the next sampling instant in a myopic manner based on knowledge learned from previous samples, and a uniform sampling scheme that uses a fixed sampling period to minimize the global cost function. The exact analytical cost functions and their respective upper bounds for both sampling schemes have been developed by using an optimum thresholding-based LSE algorithm. The cost functions and upper bounds were expressed as explicit functions of their respective design parameters, and they are minimized by choosing the next sampling instant for dynamic active sensing, or the optimum sampling rate for uniform sampling. Numerical and simulation results demonstrate that both algorithms can obtain accurate LSE with a small number of samples under a stringent power constraint,

and the uniform sampling scheme slightly outperforms the dynamic active sampling scheme.

## 2.9 Appendix of the Copyright

### 2.9.1 Copyright Clearance


**Thesis / Dissertation Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

## 2.10 References

[1] Ian F. Akyildiz, Brandon F. Lo, and Ravikumar Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Phys. Commun.*, 4(1):40–62, March 2011.

[2] Brent Bryan, Jeff Schneider, Robert Nichol, Christopher Miller, Christopher Genovese, and Larry Wasserman. Active learning for identifying function threshold boundaries. In *NIPS*. Citeseer, 2005.

[3] John W Craig. A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations. In *Military Communications Conference, 1991. MILCOM'91, Conference Record, Military Communications in a Changing World., IEEE*, pages 571–575. IEEE, 1991.

[4] Karthik Dantu and Gaurav Sukhatme. Detecting and tracking level sets of scalar fields using a robotic sensor network. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3665–3672. IEEE, 2007.

[5] Houcem Gazzah, Phillip A Regalia, and J-P Delmas. Asymptotic eigenvalue distribution of block toeplitz matrices and application to blind simo channel identification. *Information Theory, IEEE Transactions on*, 47(3):1243–1251, 2001.

[6] Alexander Goldenshluger and Assaf Zeevi. The hough transform estimator. *Annals of statistics*, pages 1908–1932, 2004.

[7] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1344–1350. AAAI Press, 2013.

[8] Robert M Gray. Toeplitz and circulant matrices: A review. *Communications and Information Theory*, 2(3):155–239, 2005.

[9] Peter Hall and Kee-Hoon Kang. Bandwidth choice for nonparametric classification. *Annals of statistics*, pages 284–306, 2005.

[10] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[11] Nicolas Maisonneuve, Matthias Stevens, and Bartek Ochab. Participatory noise pollution monitoring using mobile phones. *Info. Pol.*, 15(1,2):51–71, April 2010.

[12] David M Mason, Wolfgang Polonik, et al. Asymptotic normality of plug-in level set estimates. *The Annals of Applied Probability*, 19(3):1108–1142, 2009.

[13] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[14] Clayton Scott and Mark Davenport. Regression level set estimation via cost-sensitive classification. *Signal Processing, IEEE Transactions on*, 55(6):2752–2757, 2007.

[15] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[16] Sumana Srinivasan, Krithi Ramamritham, and Purushottam Kulkarni. Ace in the hole: Adaptive contour estimation using collaborating mobile sensors. In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pages 147–158. IEEE, 2008.

[17] Ning Sun and Jingxian Wu. Optimum sampling in spatial-temporally correlated wireless sensor networks. *EURASIP J. Wireless Commun. Networking*, 2013.

[18] RM Willett and Robert D Nowak. Minimax optimal level-set estimation. *Image Processing, IEEE Transactions on*, 16(12):2965–2979, 2007.

[19] Jingxian Wu and Ning Sun. Optimum sensor density in distortion tolerant wireless sensor networks. *IEEE Transactions on Wireless Communications*, 11:2056–2064, 2012.

[20] Jing Yang, Zuoen Wang, and Jingxian Wu. Level set estimation with dynamic sparse sensing. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 487–491. IEEE, 2014.

[21] Daniel Zwillinger. *Table of integrals, series, and products*. Elsevier, 2014.

# Chapter 3

# Optimal Energy Efficient Level Set Estimation of Spatially-Temporally Correlated Random Fields

## 3.1 Abstract

Level set estimation (LSE) is the process of classifying the region(s) that the values of an unknown function exceed a certain threshold. It has a wide range of applications such as spectrum sensing or environment monitoring. In this paper, we study the the optimal LSE of a linear random field that changes with respect to time. A linear sensor network is used to take discrete samples of the spatially-temporally correlated random field in both the space and time domain, and the sensors operate under a total power constraint. The samples are congregated at a fusion center (FC), which performs LSE of the random field by using the noisy observation of the samples. Under the Gaussian process (GP) framework, we first develop an optimal LSE algorithm that can minimize the LSE error probability. The results are then used to derive the exact LSE error probability with the assistance of frequency domain analysis. The analytical LSE error probability is expressed as an explicit function of a number of system parameters, such as the distance between two adjacent nodes, the sampling period in the time domain, the signal-to-noise ratio (SNR), and the spatial-temporal correlation of the random field. With the analytical results, we can identify the optimum node distance and sampling period that can minimize the LSE error probability.

## 3.2 Introduction

Wireless sensor networks (WSNs) have been widely used in many scientific and engineering applications, including search and rescue, disaster relief, spectrum sensing, and environment monitoring, etc. Many WSNs are designed to monitor a random event, which can be modeled as a random function in the space and time domains. It is usually difficult and costly to estimate the precise values of the random function. For many applications, it is sufficient to find out the regions over which the function values exceeds a certain threshold, and this is denoted as level set estimation (LSE). The applications of LSE include terra in mapping, spectrum sensing [1], and monitoring the contours of sunlight, water pollution, and rainfall [9].

A large number of works are devoted to the development of LSE algorithms [12]– [13]. In [11], the LSE is performed by identifying the difference between two probability densities, and the method is closely related to standard binary classifications. The binary classification based method do not consider the difference between the threshold and the actual function value, which contains salient information helpful to LSE. The methods in [4, 7, 10] are developed by studying the statistical properties of the random field. These methods do not involve an intermediate reconstruction step and it is usually hard to obtain analytical conclusions on consistency and convergence. A popular LSE method is to estimate the values of the underlying function and then thresholding at the critical value [3, 5, 12]. Such an approach is easy to implement and the consistency and convergence of the algorithms can be analyzed based on certain smooth prior assumptions. In [6, 8, 15], the domains defining the function of interest are discretized into a set of small regions to perform the LSE, and the price is the

larger estimation error introduced by the discretization. The LSE of a single spatial point over infinite continuous time domain are developed in [13]. However, it does not consider the spatial variation of the random field.

In this paper, we study the LSE of a spatially-temporally correlated random field with a linear WSN. The sensor nodes are evenly distributed on a line in the spatial domain, and they periodically sample a time-varying physical quantity, such as temperature or pollution level, under a constraint on the total power per unit area. The collected samples are transmitted to a fusion center (FC), which performs the LSE by using noisy observations of the samples. The performance of the LSE depends critically on the number of sensors in a unit area, i.e. sensor density, and the sampling rate. A higher sensor density and/or sampling rate means denser sampling of the random event, which benefit the LSE estimation. On the other hand, under the constraint of a total power per unit area, a higher sensor density and/or sampling rate means less energy per sample or a lower signal-to-noise ratio (SNR), which negatively affects the LSE performance. Therefore it is important to identify the optimum sensor density in the space domain and sampling rate in the time domain.

Under GP framework , we first propose an optimum LSE algorithm that can minimize the LSE error probability. The results are then used to derive an asymptotic LSE error probability when the size of the field and the time go to infinity. The analytical LSE error probability are explicitly expressed as a function of various system parameters, such as the node distance (inverse of node density), sampling period, SNR, and spatial and temporal covariance kernels. The asymptotically optimum node density and sampling rate can then be obtained by minimizing the asymptotic LSE error probability. Simulation results show that the asymptotic results can accurately predict the performance of practical random field

of finite size and finite time duration.

### 3.3 System Model

Consider a linear sensor network with $M$ sensor nodes evenly distributed on a line. Denote the coordinate of the $m$-th node as $s_m = ml$, for $m = 1, 2, \ldots, M$, where $l$ is the distance between two adjacent sensors. Define the two-dimensional space-time coordinate vector as $\mathbf{c} = [s, t]^T \in \mathcal{X} \times \mathcal{R}_+$, where $s$ is the space coordinate, $t$ is the time variable, $\mathcal{X}$ is the linear field and $\mathcal{R}$ is the set of real numbers. Sensors are used to measure a spatial-temporally dependent physical quantity, $x(\mathbf{c})$, such as air pressure, temperature, aggregated power level of wireless signals, or density of toxic gases, etc.

It is assumed that the *prior* distribution of $\{x(\mathbf{c})\}$ is a zero-mean Gaussian process that is wide sense stationary (WSS) in both space and time. The covariance function of $\{x(\mathbf{c})\}$ is

$$k(\mathbf{c}, \mathbf{c}') = \mathbb{E}[x(\mathbf{c})x(\mathbf{c}')] = k_s(|s - s'|) \cdot k_t(|t - t'|), \tag{3.1}$$

where $k_s(\cdot), k_t(\cdot)$ are the spatial and temporal covariance functions, respectively, and both are absolutely integrable.

Due to energy limit, the sensors take discrete-time samples of the random field. The collected discrete-time samples are transmitted to a FC. Assume uniform sampling is used. Denote the sampling instants as $t_n = nd$, for $n = 1, 2, \cdots$, where $d$ is the sampling period. It is assumed that an energy $E_0 = E_c + E$ is allocated for each sample, where the constant $E_c$ is due to hardware power consumption of the sensing operation, and $E$ is the transmission

energy of a sample. Denote $\mathbf{c}_{in} := [s_i, t_n]^T$ as the space-time coordinate of the $n$-th sample from the $i$-th sensor. The samples observed at the FC can be represented as

$$y(\mathbf{c}_{in}) = \sqrt{E}x(\mathbf{c}_{in}) + \xi(\mathbf{c}_{in}) \tag{3.2}$$

where $\xi(\mathbf{c}_{in})$ includes the effects of observation noise and channel distortions. It is assumed that $\xi(\mathbf{c}_{in})$ is a white Gaussian process with zero-mean and variance $\sigma^2$. The sensor nodes operate under the constraint of a fixed power $P_0$ per unit area. Given a sensing system with node density $\delta = \frac{1}{l}$, the energy allocated to one sample is thus $E_0 = \frac{P_0 d}{\delta} = P_0 dl$.

The FC uses the discrete-time samples to estimate the $\gamma$-level set of $\{x(\mathbf{c})\}$, which is defined as

$$\mathcal{S}(t) := \{[s,t]^T : s \in \mathcal{X}, x(s,t) > \gamma\}. \tag{3.3}$$

Without loss of generality, we assume $\gamma > 0$.

Denote the estimated level set as $\hat{\mathcal{S}}(t)$. Then the level set estimation error up to time $t$ is defined as

$$e(t) := \int_0^t \int_{\mathcal{X}} \mathbb{I}\left\{[s,\mu]^T \in \Delta(\mathcal{S}(\mu), \hat{\mathcal{S}}(\mu))\right\} ds\, d\mu \tag{3.4}$$

where $\Delta(\mathcal{S}(t), \hat{\mathcal{S}}(t)) = (\mathcal{S}(t) \cap \hat{\mathcal{S}}^c(t)) \cup (\mathcal{S}^c(t) \cap \hat{\mathcal{S}}(t))$ denotes the symmetric difference between two sets, $\mathcal{S}^c$ is the complement of $\mathcal{S}$, and $\mathbb{I}\{\mathcal{E}\} = 1$ if the event $\mathcal{E}$ is true and 0 otherwise.

Then, the LSE problem can be formulated as

$$\begin{aligned} \min. \quad & \lim_{t,M\to\infty} \frac{\delta}{Mt}\mathbb{E}[e(t)] \\ \text{s.t.} \quad & E = P_0 d/\delta - E_c \geq 0 \end{aligned} \tag{3.5}$$

In the cost function, the LSE error is normalized by the time duration $t$ and the length of the area of interest $V = M/\delta$. The optimization is performed with respect to the node density $\delta$ and the sampling rate $r = 1/d$.

### 3.4 Optimal Level Set Estimation in GP

We first study the optimal LSE for given $\delta$ and $d$ and derive the corresponding LSE error probability. The results will be used to identify the solutions to (3.5) in the next section.

Denote $\mathbf{x}_n$ and $\mathbf{y}_n \in \mathcal{R}^{M\times 1}$ be the vectors containing the true and observed data samples that the FC collects from all $M$ sensors at time $t_n$, respectively. Define $\mathbf{x}_{1:n} := \{\mathbf{x}_i\}_{i=1}^n$ and $\mathbf{y}_{1:n} := \{\mathbf{y}_i\}_{i=1}^n$ be the sets of true and observed discrete-time data from time $t_1$ to $t_n$, respectively.

Since $\{x(\mathbf{c})\}$ is Gaussian process, given $\mathbf{y}_{1:n}$, the distribution of $x(\mathbf{c})$ is still Gaussian, with conditional mean, $\hat{m}_n(\mathbf{c})$, and conditional variance, $\hat{k}_n(\mathbf{c})$, given by

$$\hat{m}_n(\mathbf{c}) = \mathbf{r}_{x\mathbf{y}_{1:n}}(\mathbf{c})\mathbf{R}^{-1}_{\mathbf{y}_{1:n}\mathbf{y}_{1:n}}\mathbf{y}_{1:n}, \tag{3.6}$$

$$\hat{k}_n(\mathbf{c}) = k(\mathbf{c},\mathbf{c}) - \mathbf{r}_{x\mathbf{y}_{1:n}}(\mathbf{c})\mathbf{R}^{-1}_{\mathbf{y}_{1:n}\mathbf{y}_{1:n}}\mathbf{r}_{x\mathbf{y}_{1:n}}(\mathbf{c})^T, \tag{3.7}$$

where $\mathbf{r}_{x\mathbf{y}_{1:n}}(\mathbf{c}) := \mathbb{E}[x(\mathbf{c})\mathbf{y}_{1:n}^T] \in \mathcal{R}^{nM\times 1}$ and $\mathbf{R}_{\mathbf{y}_{1:n}\mathbf{y}_{1:n}^T} := \mathbb{E}[\mathbf{y}_{1:n}\mathbf{y}_{1:n}^T] \in \mathcal{R}^{nM\times nM}$.

Based on (4.8), the GP regression based LSE algorithm is given in Algorithm 5.

**Algorithm 2** GP regression based level set estimation

1: Input: $\mathbf{t}_n$ and $\mathbf{y}_{1:n}$
2: Run GP regression for $\forall t \in [0, t_n], s \in \mathcal{X}$ to get $\hat{m}_n(\mathbf{c})$ with (4.8).
3: Threshold $\hat{m}_n(\mathbf{c})$:

$$\hat{\mathcal{S}}(t) = \{[s,t]^T : s \in \mathcal{X}, \hat{m}_n(\mathbf{c}) > \gamma\}, \quad 0 \le t \le t_n$$

4: Output $\hat{\mathcal{S}}(t)$.

*Theorem 3.1*: Algorithm 1 is optimal with given $\mathbf{y}_{1:n}$, i.e., it minimizes the conditional LSE error probability, $\mathbb{E}[e(t_n)|\mathbf{y}_{1:n}]$. The corresponding minimum LSE error probability is

$$\mathbb{E}[e(t_n)|\mathbf{y}_{1:n}] = \int_0^{t_n} \int_{\mathcal{X}} Q\left(\frac{|\gamma - \hat{m}_n(\mathbf{c})|}{\sqrt{\hat{k}_n(\mathbf{c})}}\right) ds\, dt \tag{3.8}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2} du$ is the Gaussian-$Q$ function.

**Proof:**

Given $\mathbf{y}_{1:n}$, the expected LSE error can be calculated as

$$\begin{aligned}
&\mathbb{E}[e(t_n)|\mathbf{y}_{1:n}] \\
&= \int_0^{t_n} \int_{\mathcal{X}} \mathbb{P}\left[ [s,t]^T \in \Delta(\mathcal{S}(t), \hat{\mathcal{S}}(t)) \middle| \mathbf{y}_{1:n} \right] ds\, dt \\
&= \int_0^{t_n} \int_{\mathcal{X}} \Big( \mathbb{P}\left[ x(\mathbf{c}) \le \gamma | \, \mathbf{y}_{1:n} \right] \cdot \mathbb{I}\{[s,t]^T \in \hat{\mathcal{S}}(t)\} \\
&\qquad + \mathbb{P}\left[ x(\mathbf{c}) > \gamma | \, \mathbf{y}_{1:n} \right] \cdot \mathbb{I}\{[s,t]^T \in \hat{\mathcal{S}}^c(t)\} \Big) ds\, dt
\end{aligned} \tag{3.9}$$

Therefore, the optimal estimator that minimizes (4.10) is to let

$$\mathbf{c} \in \begin{cases} \hat{\mathcal{S}}(t), & \text{if } \mathbb{P}\left[x(\mathbf{c}) > \gamma | \mathbf{y}_{1:n}\right] > \mathbb{P}\left[x(\mathbf{c}) \leq \gamma | \mathbf{y}_{1:n}\right] \\ \hat{\mathcal{S}}^c(t), & \text{if } \mathbb{P}\left[x(\mathbf{c}) > \gamma | \mathbf{y}_{1:n}\right] \leq \mathbb{P}\left[x(\mathbf{c}) \leq \gamma | \mathbf{y}_{1:n}\right] \end{cases} \tag{3.10}$$

for every $\mathbf{c} = [s,t]^T$ with $t \in [0, t_n]$ and $s \in \mathcal{X}$.

Since $x(\mathbf{c})$ given $\mathbf{y}_{1:n}$ is still Gaussian distributed with mean and variance given in (4.8) and (4.9), we have

$$\mathbb{P}\left[x(\mathbf{c}) > \gamma | \mathbf{y}_{1:n}\right] = Q\left(\frac{\gamma - \hat{m}_n(\mathbf{c})}{\sqrt{\hat{k}_n(\mathbf{c})}}\right) \tag{3.11}$$

The optimal estimator defined in (4.11) is then reduced to compare $\hat{m}_n(\mathbf{c})$ with $\gamma$. If $\hat{m}_n(\mathbf{c}) > \gamma$, the probability in (4.12) is greater than 1/2, thus, we should let $\mathbf{c} \in \hat{\mathcal{S}}(t_n)$; otherwise, we let $\mathbf{c} \in \hat{\mathcal{S}}^c(t_n)$. A combination of (4.10) and (4.12) results in (4.13). ■

### 3.5 Optimal Uniform Sampling

In this section, we will first find the unconditional LSE error probability using the results from Algorithm 5. The analytical results will then be used to identify the optimum sampling period and node density that can minimize the LSE error probability.

The cost function in (3.5) is the time and space averaged unconditional error probability. The unconditional LSE error probability can be alternatively expressed as

$$\mathbb{E}[e(t_n)] = \mathbb{E}_{\mathbf{y}_{1:n}}\left\{\mathbb{E}[e(t_n)|\mathbf{y}_{1:n}]\right\} \tag{3.12}$$

From (4.13) and (3.12), the cost function depends on the posterior mean $\hat{m}_n(\mathbf{c})$ and variance $\hat{k}_n(\mathbf{c})$. From (4.8) and (4.9), we have

$$\hat{m}_n(\mathbf{c}) = \sqrt{E}\mathbf{r}_{x\mathbf{x}_{1:n}}(\mathbf{c}) \left(E\,\mathbf{R}_{\mathbf{x}_{1:n}\mathbf{x}_{1:n}} + \sigma^2\mathbf{I}_{nM}\right)^{-1} \mathbf{y}_{1:n} \tag{3.13}$$

$$\hat{k}_n(\mathbf{c}) = k(\mathbf{c},\mathbf{c}) - E\,\mathbf{r}_{x\mathbf{x}_{1:n}}(\mathbf{c}) \left(E\,\mathbf{R}_{\mathbf{x}_{1:n}\mathbf{x}_{1:n}} + \sigma^2\mathbf{I}_{nM}\right)^{-1} \mathbf{r}^T_{x\mathbf{x}_{1:n}}(\mathbf{c}). \tag{3.14}$$

where $\mathbf{r}_{x\mathbf{x}_{1:n}}(\mathbf{c}) = \mathbb{E}\left[x(\mathbf{c})\mathbf{x}^T_{1:n}\right] \in \mathcal{R}^{1\times nM}$ and $\mathbf{R}_{\mathbf{x}_{1:n}\mathbf{x}_{1:n}} = \mathbb{E}[\mathbf{x}_{1:n}\mathbf{x}^T_{1:n}] \in \mathcal{R}^{nM\times nM}$.

The posterior mean is a function of $\mathbf{y}_{1:n}$, whereas the posterior variance is a constant independent of $\mathbf{y}_{1:n}$. Since $\mathbf{y}_{1:n}$ is zero-mean Gaussian distributed, it can be easily shown that $\hat{m}_n(\mathbf{c})$ is zero-mean Gaussian distributed with variance

$$\sigma^2_{\hat{m}_n}(\mathbf{c}) = E\,\mathbf{r}_{x\mathbf{x}_{1:n}}(\mathbf{c}) \left(E\,\mathbf{R}_{\mathbf{x}_{1:n}\mathbf{x}_{1:n}} + \sigma^2\mathbf{I}_{nM}\right)^{-1} \mathbf{r}^T_{x\mathbf{x}_{1:n}}(\mathbf{c}) \tag{3.15}$$

From (3.14) and (3.15), we have $\hat{k}_n(\mathbf{c}) = k(\mathbf{c},\mathbf{c}) - \sigma^2_{\hat{m}_n}(\mathbf{c})$.

The variance in (3.15) depends on a number of factors, such as the spatial and temporal covariance, the sampling period $d$, the node density $\delta$, the energy per sample $E$, the number of sensors $M$ and the time instant $t$. As $n \to \infty$ and $M \to \infty$, we have the following asymptotic results of $\sigma^2_{\hat{m}_n}(\mathbf{c})$ and $\hat{k}_n(\mathbf{c})$.

*Theorem 3.2*: Define the asymptotic posterior variance $\sigma^2_e(u,v) := \lim_{n,M\to\infty} \hat{k}_n(\mathbf{c})$, where $u = \frac{s}{l} - \lfloor\frac{s}{l}\rfloor \in [0,1]$ is the relative position of $s$ between two adjacent samples in space and $v = \frac{t}{d} - \lfloor\frac{t}{d}\rfloor \in [0,1]$ is the relative position of $t$ between two adjacent samples in time. We

have

$$\sigma_e^2(u,v) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left[ \Psi_{00}(f_s,f_t) - \frac{|\Psi_{uv}(f_s,f_t)|^2}{\Psi_{00}(f_s,f_t) + \frac{1}{\gamma_0(d/\delta - d_c)}} \right] df_s\, df_t \tag{3.16}$$

where $\gamma_0 := \frac{P_0}{\sigma^2}$ is the signal-to-noise ratio (SNR), $d_c := \frac{E_c}{P_0}$ is hardware energy normalized by the average power constraint, and

$$\Psi_{uv}(f_s,f_t) := \sum_{m=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} k_s((m+u)l) k_t((n+v)d) e^{-2\pi j(mf_s + nf_t)} \tag{3.17}$$

is the 2-D discrete-time Fourier transform (DTFT) of sequences $\left\{k_s((m+u)l)k_t((n+v)d)\right\}_{m,n}$.

In addition,

$$\lim_{n,M\to\infty} \sigma^2_{\hat{m}_n}(\mathbf{c}) = 1 - \sigma_e^2(u,v). \tag{3.18}$$

**Proof:** To simplify notation, denote $x_{i+u,n+v} = x([(i+u)l,(n+v)d]^T)$ and $y_{i,n} = y([il,nd]^T)$. The linear minimum mean squared error (LMMSE) estimation of $x_{i+u,n+v}$ based on $\{y_{i,n}\}_{i,n}$ is

$$\hat{x}_{i+u,n+v} = \sum_{m=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} h(m,k) y_{i-m,n-k} \tag{3.19}$$

where $\{h(m,k)\}$ is the impulse response of the LMMSE filter.

Based on the orthogonal principle, $\mathbb{E}[(x_{i+u,n+v} - \hat{x}_{i+u,n+v})y_{i',n'}] = 0$, we have

$$\sum_{m=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} h(m,k)r_{yy}(i-m,n-k) = r_{xy}(i+u,n+v) \tag{3.20}$$

where

$$r_{yy}(i,n) = \mathbb{E}[y_{i'+i,n'+n}y_{i',n'}] = E\,k_s(il)k_t(nd) + \sigma^2 \tag{3.21}$$

$$\begin{aligned} r_{xy}(i+u,n+v) &= \mathbb{E}[x_{i'+i+u,n'+n+v}y_{i',n'}] \\ &= \sqrt{E}k_s((i+u)l)k_t((n+v)d) \end{aligned} \tag{3.22}$$

Based on the convolution theorem [14], converting (3.20)-(3.22) into the frequency domain with 2-D DTFT, we have

$$H(f_s,f_t) = \frac{\sqrt{E}\Psi_{uv}(f_s,f_t)}{E\,\Psi_{00}(f_s,f_t) + \sigma^2} \tag{3.23}$$

Based on the orthogonal principle, the MSE can be calculated as

$$\sigma_e^2(u,v) = \mathbb{E}[(x_{i+u,n+v} - \hat{x}_{i+u,n+v})x_{i+u,n+v}] \tag{3.24}$$

Combining (3.19) and (3.24) yields

$$\begin{aligned}
&\sigma_e^2(u,v) = k_s(0)k_t(0) - \\
&\sqrt{E}\sum_{m=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} h(m,k)k_s(-(m+u)l)k_t(-(k+v)d) \\
&= \int_{-\frac{1}{2}}^{\frac{1}{2}}\int_{-\frac{1}{2}}^{\frac{1}{2}} [\Psi_{00}(f_s,f_t) - H(f_s,f_t)\Psi_{uv}^*(f_s,f_t)]\, df_s df_t
\end{aligned} \tag{3.25}$$

Then (3.16) can be obtained by combining the above equation with (3.23). ■

From Theorem 3.2, we can see that as $n, M \to \infty$, $\hat{m}_n(il+u, kd+v)$ is a zero-mean Gaussian random variable with variance $1-\sigma_e^2(u,v)$. Therefore, as $n, M \to \infty$, the statistical properties of $\hat{m}_n(\mathbf{c})$ are periodic in space and time with periods $l$ and $d$, respectively. Define $\hat{m}(u,v) := \lim_{n,M\to\infty} \hat{m}_n([(i+u)l, (k+v)d]^T)$. We have the following corollary regarding the distribution of $\hat{m}(u,v)$.

*Corollary 3.1*: As $n, M \to \infty$, $\hat{m}(u,v)$ is zero-mean Gaussian distributed with variance $1-\sigma_e^2(u,v)$, that is, $\hat{m}(u,v) \sim \mathcal{N}(0, 1-\sigma_e^2(u,v))$.

With the asymptotic results in Theorem 3.2 and Corollary 3.1, we can get an explicit expression of the cost function in (3.5), and the result is given in the following theorem.

*Theorem 3.3*: The cost function in (3.5), $\mathcal{J}(d,l) := \lim_{n,M\to\infty} \frac{1}{nd\cdot Ml}\mathbb{E}[e(nd)]$, can be expressed as

$$\begin{aligned}
\mathcal{J}(d,l) = \frac{1}{\pi}\int_0^1\int_0^1\int_0^{\pi/2} &\left[1 + \frac{1-\sigma_e^2(u,v)}{\sigma_e^2(u,v)\sin^2(\theta)}\right]^{-1/2} \\
&\times \exp\left(-\frac{\gamma^2/2}{1-\sigma_e^2(u,v)\cos^2(\theta)}\right) du\, dv\, d\theta.
\end{aligned} \tag{3.26}$$

**Proof:** We first consider the conditional cost function, $\lim_{n,M\to\infty}\frac{1}{nd\cdot Ml}\mathbb{E}[e(nd)|\mathbf{y}_{1:n}]$. From (4.13)-(3.12), we have

$$\mathbb{E}[e(nd)|\mathbf{y}_{1:n}] = \sum_{i=1}^{M}\sum_{k=1}^{n}\int_{(i-1)l}^{il}\int_{(k-1)d}^{kd} Q\left(\frac{|\gamma-\hat{m}_n(\mathbf{c})|}{\sqrt{\hat{k}_n(\mathbf{c})}}\right) ds\, dt$$

Performing change of variable, $s=(i-1)l+ul$, and $t=(k-1)d+vd$ in the above integral , and using the results from Theorem 3.2, we have

$$\lim_{n,M\to\infty}\frac{1}{nd\cdot Ml}\mathbb{E}[e(\hat{\mathcal{S}}(nd))|\mathbf{y}_{1:n}] = \int_0^1\int_0^1 Q\left(\frac{|\gamma-\hat{m}(u,v)|}{\sqrt{\sigma_e^2(u,v)}}\right) du\, dv \tag{3.27}$$

In the conditional cost function in (3.27), there is only one random variable, $\hat{m}(u,v)\sim\mathcal{N}(0,1-\sigma_e^2(u,v))$, which is a function of $\mathbf{y}_{1:n}$. Therefore the unconditional cost function can be expressed as

$$\mathcal{J}(d,l)=\int_0^1\int_0^1\mathbb{E}_{\hat{m}(u,v)}\left[Q\left(\frac{|\gamma-\hat{m}(u,v)|}{\sqrt{\sigma_e^2(u,v)}}\right)\right] du\, dv \tag{3.28}$$

With Craig's alternative expression of the $Q$-function [2], (3.28) can be reformulated as

$$\mathcal{J}(d,l)=\frac{1}{\pi}\int_0^1\int_0^1\int_0^{\frac{\pi}{2}}\mathbb{E}_{\hat{m}(u,v)}\left[e^{-\frac{(\hat{m}(u,v)-\gamma)^2}{2\sigma_e^2(u,v)\sin^2(\theta)}}\right] du\, dv d\theta \tag{3.29}$$

Define $Z:=\frac{(\hat{m}(u,v)-\gamma)^2}{1-\sigma_e^2(u,v)}$, which is a non-central $\chi^2$-distributed random variable with one

degree-of-freedom and the non-centrality parameter $\frac{\gamma^2}{1-\sigma_{\tilde{e}}^2(u,v)}$. The moment generating function (MGF) of $Z$, $M_Z(s) = \mathbb{E}_Z[e^{sZ}]$, is

$$M_Z(s) = \exp\left[\frac{s}{(1-2s)}\frac{\gamma^2}{1-\sigma_e^2(u,v)}\right]\frac{1}{\sqrt{1-2s}}. \tag{3.30}$$

Combining (3.29) with (3.30) yields (3.26). ■

Theorem 3.3 gives the exact analytical expression of the cost function, which is expressed as a function of $d$, $\delta$ and other system parameters such as the SNR $\gamma_0$, the temporal and spatial correlation covariance function $k_t(x)$ and $k_s(x)$, and the normalized hardware energy consumption $d_c$. Thus, given $\{\gamma_0, \rho, d_c, k_t, k_s\}$, we can identify $d$ and $\delta$ that minimizes the cost function $\mathcal{J}$ in (3.26). The integration limits in (3.26) are finite, thus the integrals in (3.26) can be evaluated with numerical integration with high precision.

In the special case that the covariance functions are the power law kernels, i.e., $k_s(x) = \rho_s^{|x|}$ and $k_t(x) = \rho_t^{|x|}$ with $\rho_s$ and $\rho_t$ being the power law coefficients in the spatial and temporal domains, respectively, we can get the closed-form expressions of the 2-D DTFT in (3.17)

$$\begin{aligned}\Psi_{uv}(f_s, f_t) = &\frac{\rho_t^{-vd}[\rho_t^d(1-\rho_t^{2vd})e^{2\pi j f_t} + \rho_t^{2vd} - \rho_t^{2d}]}{1+\rho_t^{2d} - 2\rho_t^d\cos(2\pi f_t)} \\ &\times \frac{\rho_s^{-ul}[\rho_s^l(1-\rho_s^{2ul})e^{2\pi j f_s} + \rho_s^{2ul} - \rho_s^{2l}]}{1+\rho_s^{2l} - 2\rho_s^l\cos(2\pi f_s)}\end{aligned} \tag{3.31}$$

Numerical results show that the cost function with the power law covariance is convex in both sampling period $d$ and node distance $l$, thus we can always identify the optimal $d$ and $l$ by using the Karush-Kuhn-Tucker (KKT) conditions.

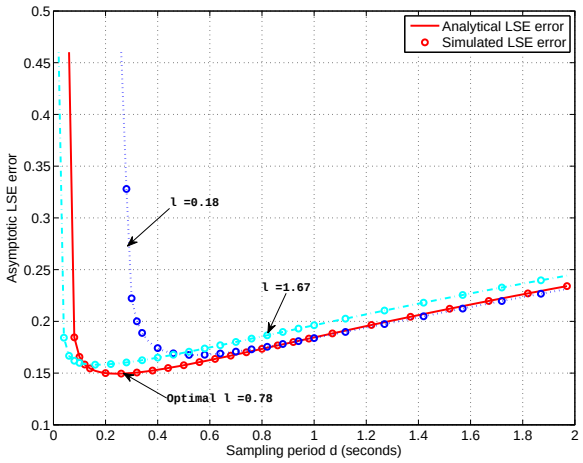



Figure 3.1: Asymptotic LSE error probabilities as a function of sampling period $d$ under various node distances $l$.

### 3.6 Simulation Results

Simulation results are presented in this section to demonstrate the performance of the proposed level set estimation algorithm with power law kernels. The threshold of the level set $\gamma$ is set to be 0.1.

Fig. 3.1 shows the asymptotic LSE error probability as a function of the sampling period $d$ under different node distances $l$. The SNR is $\gamma_0 = 10$ dB. The temporal and spatial power law coefficients are set to be $\rho_t = 0.5$ and $\rho_s = 0.8$, respectively. The normalized hardware energy is $d_c = 0.05$. The simulation results are obtained with $n = 100$ and $M = 100$. The simulation results with finite $n$ and $M$ matches very well with the numerically analytical results with infinite $n$ and $M$. Thus the asymptotic analytical results give a very good approximation of the performance of practical systems with finite $n$ and $M$. Given $l$, the asymptotic LSE error probability is a convex function of sampling period $d$ for $d > d_c/l$. When $d = d_c/l$, all energy is consumed by the sensing operation and there is no energy to transmit information to FC, which leads to a maximum LSE error, $Q(|\gamma|) = 0.46$, the same

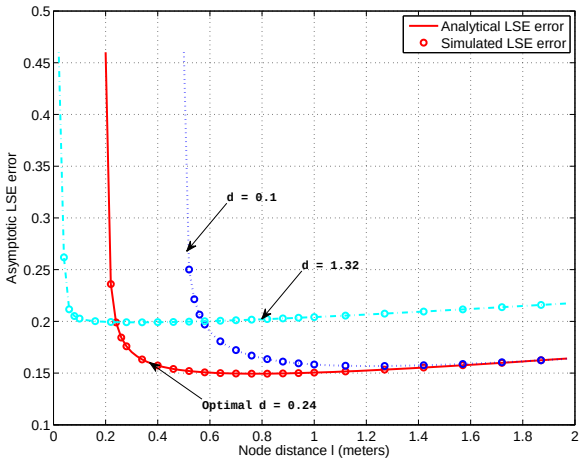



Figure 3.2: Asymptotic LSE error probabilities as a function of node distances $l$ under various sampling periods $d$.

error with a random decision estimator. The minimum LSE error probability is achieved with the optimal $l^* = 0.78$, which corresponds to an LSE error probability of 0.149 .

Similarly, Fig. 3.2 presents the asymptotic LSE error probability as a function of node distance $l$ with different sampling periods $d$. All other parameters are the same as Fig. 3.1. Given $d$, the asymptotic LSE error probability is also a convex function of node distance $l$ for $l > d_c/d$. The special case of $l = d_c/d$ results in a maximum LSE error, $Q(|\gamma|) = 0.46$. The minimum LSE error probability 0.149 is achieved with $d^* = 0.24$ and $l^* = 0.78$.

Fig. 3.3 shows the optimal asymptotic LSE error probability as a function of the SNR, under different values of $d_c$. The power law coefficients are $\rho_t = 0.5$ and $\rho_s = 0.8$. The optimal asymptotic LSE error probability is obtained by identifying the optimal values of $d$ and $l$ for each configuration. As expected, the optimal asymptotic LSE error probability is a decreasing function of SNR. For a given SNR, a larger $d_c$ yields a larger optimal asymptotic LSE error probability, due to the fact that more energy are consumed by the hardware.

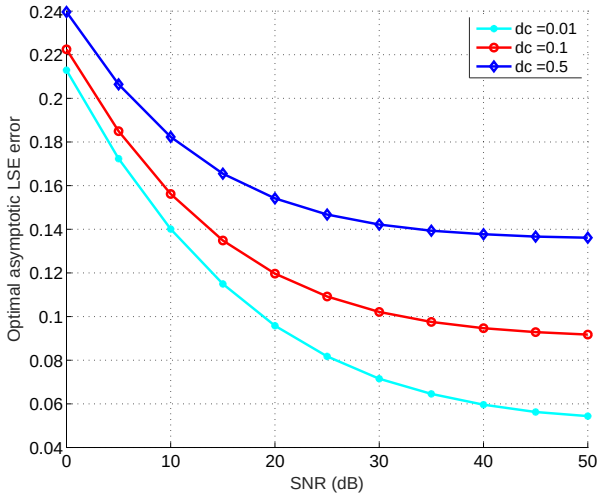


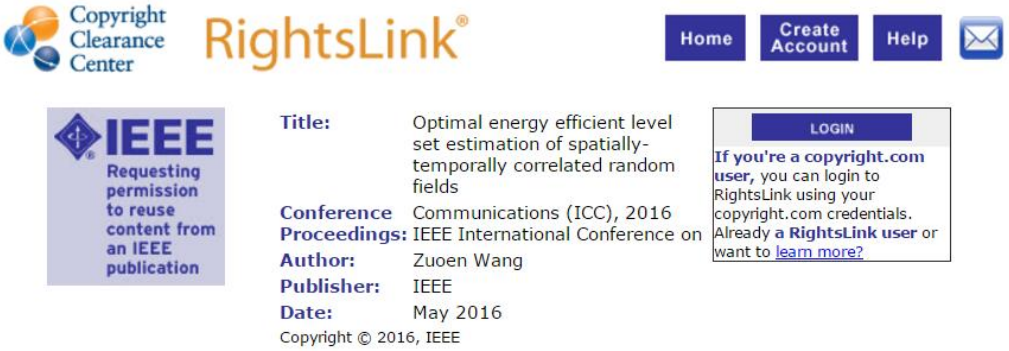

Figure 3.3: Optimal asymptotic LSE error probabilities as a function of SNR under various $d_c$ ($\rho = 0.5$).

## 3.7 Conclusions

We have studied the optimal level set estimation of a temporally-spatially correlated random field with linear sensor network, under a total power constraint. The optimal LSE algorithm was developed by using regression of a 2-D Gaussian process. Then the exact LSE error probability was derived with the assistance of frequency domain analysis. The LSE error probability has been expressed as an explicit function of a number of system parameters, such as the node distance in the space domain, the sampling period in the time domain, the covariance kernel functions, and the SNR, etc. The optimum node distance and sampling period can then be identified to minimize the LSE error probability. Simulation results demonstrated that the proposed algorithm can achieve accurate and efficient LSE.

## 3.8 Appendix of the Copyright

### 3.8.1 Copyright Clearance




**Thesis / Dissertation Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

### 3.9 References

[1] Ian F. Akyildiz, Brandon F. Lo, and Ravikumar Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Phys. Commun.*, 4(1):40–62, March 2011.

[2] John W Craig. A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations. In *Military Communications Conference, 1991. MILCOM'91, Conference Record, Military Communications in a Changing World., IEEE*, pages 571–575. IEEE, 1991.

[3] Karthik Dantu and Gaurav Sukhatme. Detecting and tracking level sets of scalar fields using a robotic sensor network. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3665–3672. IEEE, 2007.

[4] Alexander Goldenshluger and Assaf Zeevi. The hough transform estimator. *Annals of statistics*, pages 1908–1932, 2004.

[5] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1344–1350. AAAI Press, 2013.

[6] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 265–272. ACM, 2005.

[7] Peter Hall and Kee-Hoon Kang. Bandwidth choice for nonparametric classification. *Annals of statistics*, pages 284–306, 2005.

[8] Yifei Ma, Roman Garnett, and Jeff Schneider. Active area search via bayesian quadrature. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 595–603, 2014.

[9] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[10] David M Mason, Wolfgang Polonik, et al. Asymptotic normality of plug-in level set estimates. *The Annals of Applied Probability*, 19(3):1108–1142, 2009.

[11] Mario Roederer and Richard R Hardy. Frequency difference gating: a multivariate method for identifying subsets that differ between samples. *Cytometry*, 45(1):56–64, 2001.

[12] Clayton Scott and Mark Davenport. Regression level set estimation via cost-sensitive classification. *Signal Processing, IEEE Transactions on*, 55(6):2752–2757, 2007.

[13] Zuoen Wang, Jingxian Wu, Jing Yang, and Hai Lin. Energy-efficient wireless sensing for level set estimations. *Access, IEEE*, 3:1480–1490, 2015.

[14] Eric W Weisstein. Convolution theorem. *From MathWorld-A Wolfram Web Resource, 2006a. URL http://mathworld. wolfram. com/ConvolutionTheorem. html*, 2014.

[15] Jing Yang, Zuoen Wang, and Jingxian Wu. Level set estimation with dynamic sparse sensing. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 487–491. IEEE, 2014.

# Chapter 4

# Level Set Estimation of Spatial-temporally Correlated Random Fields with Active Sparse Sensing

## 4.1 Abstract

In this paper, we study the level set estimation of a spatial-temporally correlated random field by using a small number of spatially distributed sensors. The level sets of a random field are defined as regions where data values exceed a certain threshold. The identification of the boundaries of such sets is an important theoretical problem with a wide range of applications such as spectrum sensing, urban sensing, and environmental monitoring, etc. We propose a new active sparse sensing and inference scheme, which can achieve rapid and accurate extraction of level sets in a large random field by using a small number of data samples strategically and sparsely selected from the field. A Gaussian process (GP) prior model is used to capture the spatial-temporal correlations inherent in the random field. It is first shown that the optimal level set estimation can be achieved by performing a GP regression with all data samples and then thresholding the regression results. We then investigate the active sparse sensing scheme, where a central controller dynamically selects a small number of sensing locations according to the information revealed from past measurements, with the objective to minimize the expected level set estimation error probability. The expected estimation error probability is explicitly expressed as a function of the selected sensing locations, and the results are used to formulate the optimal sensing location selection problem

as a combinatorial problem. Two low complexity greedy algorithms are developed by using analytical upper bounds of the expected estimation error probability. Both simulation and experiment results demonstrate that the greedy algorithms can achieve significant performance gains over baseline passive sensing algorithms and the GP Upper Confidence Bound (GP-UCB) level set estimation algorithm.

## 4.2 Introduction

Large-scale sensing has played a critical role in many scientific and engineering fields, such as spectrum sensing and environment monitoring, etc. For many large-scale sensing applications efficient level set identification is a crucial task. Level set estimation is the process of using observations of a function $f(s)$ defined on a Hilbert space $\mathcal{X}$ to estimate the region(s) in $\mathcal{X}$ where the function value exceeds some critical value $\gamma$; i.e. $\mathcal{S} := \{\mathbf{s} \in \mathcal{X} : f(\mathbf{s}) \geq \gamma\}$. Level set estimation is of paramount importance in many large-scale sensing applications, including the following examples.

- Spectrum sensing in cognitive radio networks [1]: fast identification of the boundary of "spectrum holes" in space and frequency domains is crucial for the construction of spectrum map [8], a dynamic database providing real-time information and predictions on spectrum usage at a given area over a wide range.

- Urban sensing: accurate monitoring and tracking of the range of a widespread phenomena, such as traffic congestion [19], air/water/noise pollution [18], damages caused by hurricane, is of critical importance.

- Environment monitoring: contours of sunlight, rainfall and other key environmental

factors are critical for the understanding and tracking of biosystem ecology [17].

- Swarming sensing: to identify urban tomography for military operations by swarming coordination of multiple unmanned aerial vehicles (UAVs) for collaborative sensing [23].

In these and many other applications identifying level sets is the primary task, while estimating the value of the function (i.e. the power in spectrum sensing) away from the level set boundary is often secondary if not irrelevant. Consequently level set estimation can be equivalently considered as a mapping problem that draws the level contour or boundary in a random field. Intuitively data that are further away from the boundary are usually quite distinct from the level of interests, thus there is less ambiguity in terms of level set identification in those regions. Therefore it is desirable to collect more data samples or place more sensors at the locations where the boundary is likely to lie.

This paper describes a new active sparse sensing and inference scheme for rapid and accurate extraction of level sets of a spatial-temporally correlated random field. One of the main novelties of the proposed scheme is that it can dynamically adjust the sensing locations through active learning and adaptation of level set boundaries by analyzing past sensing data. Therefore the proposed scheme can achieve accurate estimation of the level sets with only a small number of sensors strategically placed at critical locations of the random field.

While many methods have been devised for level set estimation in a static setting [9, 11–13, 27], the temporally evolving nature of the random field requires a dynamic level set estimation, which makes the estimation problem different and challenging. Besides, existing work in this area often assumes that the measurements and the sensor locations are static, as opposed to dynamically selected [2, 21, 35]. How to actively sense the field for a fast and

accurate level set estimation has been rarely investigated [32, 33, 36]. On the other hand, active learning and its applications have been extensively investigated in the machine learning community [28]. The common goal of active learning algorithms is to adaptively select statistically optimal training data with information gleaned from previous observations [5]. Numerous sample selection criteria have been proposed [16]. The active learning approach has been widely applied in sensing networks for mobile path planning [15, 29], and sensor placement [14], etc. The active sensing approach we propose in this paper inherits the essence of active learning. The problem studied in this paper is fundamentally distinct from these works in two ways; the first is that our objective is to estimate the level set instead of estimating the function values and the second is the time varying nature of our problem.

We introduce a Gaussian process (GP) prior model to capture the spatial-temporal correlations inherent in the random field [4, 7, 25, 26]. GPs have been exploited to address the sensor location selection problem in static sensor networks [14, 15], and the sensing path planning problem in mobile sensor networks [3], etc. It is pointed out in [22] that the actual multivariate distribution underlying a set of data is difficult to obtain. Given this uncertainty, the multivariate Gaussian distribution is a natural assumption because it is the distribution of maximum entropy when all that is known is the mean and covariance matrix. According to [34], non-Gaussian data can often be made approximately Gaussian by transformation to a new scale (e.g. by taking logarithms or square-roots), and this is widely followed as the best practice in the analysis of soil data.

The objectives of many sensing applications are often to minimize the uncertainty of the posterior distribution of the function under certain constraints. The level set estimation problem studied in this paper is fundamentally different from those formulations, due to

the fact that our objective is not to reconstruct the entire function with minimum expected mean squared error (MSE), but to estimate the level set of the underlying function accurately. Similar level set estimation problems under a GP formulation are studied in [9, 30] by employing a Gaussian Process Upper Confidence Bound (GP-UCB) algorithm. The GP-UCB algorithm selects the sensing locations by minimizing a cost function that penalizes both the posterior variance and deviation from the level set boundary. This cost function is intuitive given that more sensors should be placed to the location of interests, that is, the level set boundaries, but it is also a surrogate objective function because this cost function does not directly measure the error probability of level set estimation. In this paper we will perform active sensing by directly minimize the level set estimation error probability. Active sensing to minimize the level set estimation error is usually a difficult task, because it requires the error probability as a function of potential future sensing locations, and the observations at those potential locations have not yet been revealed to the fusion center (FC) during the error probability analysis. In addition, the works in [9, 30] consider only spatial correlation. The temporal correlation is considered in [31], where a myopic active sensing scheme is performed in the time domain by selecting the next sampling instant to minimize the cost function accumulated up to the next sampling instant. However, the results in [31] show that the performance of time domain myopic sensing is worse than that of passive uniform sensing due to the myopic nature of the scheduling scheme. In this paper we take into consideration of both spatial and temporal correlation into the formulation and develop optimum active sensing schemes that can achieve significant performance gains over passive sensing.

Under the GP framework, we first show that the optimal level set estimation can be

achieved by performing a GP regression with all data samples and then thresholding the regression results. We then investigate the active sensing scheme, where the central controller actively selects the sensing locations according to the information gleaned from past measurements, with the objective to minimize the expected level set estimation error. Extracting information embedded in past sensed data leads to an improved estimation performance, due to the temporal correlation in the sensed signal. Meanwhile previous observations also provide some "prior" information for current sensing, which enables a more efficient sensing scheme. Intuitively to minimize the level set estimation error, sensing locations should be selected from where the boundary is likely to lie, with "prior" information gleaned from previous observations. The expected estimation error is explicitly characterized as a function of past sensing results and the potential future sensing locations, and the results are used to formulate the optimal sensing location selection problem as a combinatorial problem. Two low complexity greedy algorithms are then proposed by developing upper bounds of the expected estimation error.

Our contributions are three-fold:

1. We propose a set of new active sensing algorithms that directly minimize the level set estimation error probability, which is expressed as an explicit function of past observations and future potential sensing locations. Such a problem formulation results in new sensing algorithms that outperform existing level set algorithms that employ intuitive but surrogate cost functions.

2. We introduce a two-step active sensing scheme, where the first step is to obtain an initial estimation based on historical data samples, and the second step is to actively

probe the field to refine the initial estimation. The sensing scheme is designed to minimize the expected estimation error under a sensing budget constraint. The estimation error metric driven sensing location selection algorithm is novel. The proposed greedy algorithms are practical and efficient.

3. The problem formulation and methodology developed in this paper can benefit many large-scale sensing applications with "big data". It can also be applied to perform "information distillation", the process that extracts useful data from an ocean of data that have already been collected.

The proposed algorithms can be applied to different application scenarios. For a static wireless sensor network with a large number of sensors, we can use the algorithms to activate only a small subset of sensors at any given moment to reduce the energy consumption and prolong the life time of the entire network. For mobile networks such as UAV swarms the algorithms can be used for mobile path planning by considering additional mobility constraints. The results in this paper are developed without considering the mobility constraints, and the analytical performance results can serve as lower bounds for systems with mobility constraints.

### 4.3 System Model

We consider a sensing system with multiple sensor nodes placed over a measurement field $\mathcal{X} \subset \mathbb{R}^2$. Define the three-dimensional (3D) space-time coordinate vector as $\mathbf{c} = [\mathbf{s}, t]^T \in \mathcal{X} \times \mathbb{R}_+$, where $\mathbf{s} = [s_1, s_2]^T$ is the space coordinate, and $t$ is the time variable. Sensor nodes measure a spatial-temporally dependent physical quantity, $f(\mathbf{c})$, such as the temperature,

humidity, aggregated power level of wireless signals, or sunlight intensity, etc. The two-dimensional (2D) space coordinate is assumed here. It should be noted that all analysis and algorithm presented in this paper can also be directly extended to three-dimensional spatial cases.

The sensing samples from individual sensors are transmitted to a FC for processing. We model the sensing samples recovered at the FC as the sum of the ground truth $f(\mathbf{c})$, and a noise term $\xi$, i.e.,

$$y(\mathbf{c}) = f(\mathbf{c}) + \xi.$$

Here we use $\xi$ to capture the distortions introduced during the sensing stage as well as the transmission stage.

*Assumptions 4.1*: We make the following assumptions:

a) The *prior* distribution of $\{f(\mathbf{c})\}$ is a zero-mean Gaussian process, i.e., for any two points $\mathbf{c}_i$, $\mathbf{c}_j$, $f(\mathbf{c}_i), f(\mathbf{c}_j)$ are jointly zero-mean Gaussian distributed, with covariance $k(\mathbf{c}_i, \mathbf{c}_j)$. Any non-zero mean GP can be converted to a zero-mean GP by subtracting the original process with its mean, which can be easily estimated.

b) The Gaussian process is wide sense stationary in both space and time, and

$$k(\mathbf{c}_i, \mathbf{c}_j) = k_s(\|\mathbf{s}_i - \mathbf{s}_j\|) \cdot k_t(|t_i - t_j|), \tag{4.1}$$

where $k_s(\cdot), k_t(\cdot)$ are defined as the spatial and temporal covariance, respectively. The

$\ell_2$-norm $\|\mathbf{s}_i - \mathbf{s}_j\|$ measures the Euclidean distance between the two points with coordinates $\mathbf{s}_i, \mathbf{s}_j \in \mathcal{X}$.

c) $\xi$ is an independent Gaussian random variable with zero mean and variance $\sigma^2$, i.e., $\xi \sim \mathcal{N}(0, \sigma^2)$.

At time $t$, we are interested in identifying the $\gamma$-level set of $\{f([\mathbf{s}, t]^T)\}$, which is defined as

$$\mathcal{S}(t) := \{\mathbf{s} \in \mathcal{X} : f([\mathbf{s}, t]^T) > \gamma\} \tag{4.2}$$

Without loss of generality, we assume $\gamma > 0$.

To simplify the design and analysis, we partition the measurement field $\mathcal{X}$ with equal-sized grids. We assume that the edge length of the grid is small enough such that the signal remains approximately unchanged within one grid. Index the coordinates of the grid as $1, 2, \ldots, L$, and let $\mathbf{s}_i$ be the coordinate for the $i$-th grid point. Then, $\mathcal{X}$ can be slightly modified as $\mathcal{X} := \{\mathbf{s}_i : i = 1, \ldots, L\}$.

Similarly, the time axis is partitioned into discrete time slots, $t_1, t_2, \cdots$, where the signal stays constant in a slot, but evolves from slot to slot based on the temporal correlation of the time-varying random process. During each slot, sensing samples are collected from a number of locations. Let $\mathcal{C}_1, \mathcal{C}_2, \ldots$ be the sets of spatial coordinates the FC has collected samples from at time $t_1, t_2, \ldots$. Denote $\mathbf{f}_n$ and $\mathbf{y}_n$ as the vectors containing the true and observed data samples at $\mathcal{C}_n$, respectively. Define $\mathbf{f}^{1:n} := \{\mathbf{f}_i\}_{i=1}^n$ and $\mathbf{y}^{1:n} := \{\mathbf{y}_i\}_{i=1}^n$. Then, at the end of time slot $t_n$, the FC estimates the $\gamma$-level set $\mathcal{S}_n = \mathcal{S}(t_n)$ given $\mathcal{C}^{1:n} := \{\mathcal{C}_i\}_{i=1}^n$

and $\mathbf{y}^{1:n}$.

The optimum level set estimation algorithm should minimize the level set estimation error probability, which can be calculated at time $t_n$ as

$$e(\hat{\mathcal{S}}_n) := \frac{1}{L}\sum_{i=1}^{L}\mathbb{I}\left\{\mathbf{s}_i \in \Delta(\mathcal{S}_n, \hat{\mathcal{S}}_n)\right\} \tag{4.3}$$

where $\hat{\mathcal{S}}_n$ is the estimated level set, $\Delta(\mathcal{S}_n, \hat{\mathcal{S}}_n) = (\mathcal{S}_n \cap \hat{\mathcal{S}}_n^c) \cup (\mathcal{S}_n^c \cap \hat{\mathcal{S}}_n)$ denotes the symmetric difference between the true and estimated level set, $L$ is the total number of spatial grids, $\mathcal{S}^c$ is the complement of $\mathcal{S}$, and $\mathbb{I}\{E\} = 1$ if the event $E$ is true and 0 otherwise. In (4.3), the level set estimation error probability measures the percentage of spatial grids in which the estimated level set does not equal to the true level set.

Then at each time slot $t_n$ the dynamic level set estimation problem is to select the set of sensing locations, $\mathcal{C}_n$, based on the knowledge of $\mathcal{C}^{1:n-1}$ and $\mathbf{y}^{1:n-1}$, such that the expected estimation error $\mathbb{E}[e(\hat{\mathcal{S}}_n)]$ is minimized. Assume the sensing cost is proportional to the number of sensing actions performed by the sensor nodes. Thus at each time slot it is assumed that the system can pick up to $N$ sensing locations due to a sensing budget constraint, that is $|\mathcal{C}_n| \leq N$, for all $n$. The problem can be formulated as follows.

$$\begin{aligned} \text{minimize}_{\mathcal{C}_n} \quad & \mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n-1}, \mathcal{C}_n] \\ \text{s.t.} \quad & |\mathcal{C}_n| \leq N \end{aligned} \tag{4.4}$$

where the expectation in the cost function is performed over the GP and the noise. The cost function, $\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n-1}, \mathcal{C}_n]$, depends on both past observations, $\mathbf{y}^{1:n-1}$, and potential future

sensing locations, $\mathcal{C}_n$. Here when observation history $\mathbf{y}^{1:n-1}$ (or $\mathbf{y}^{1:n}$) is given, the sensing locations information, $\mathcal{C}^{1:n-1}$ (or $\mathcal{C}^{1:n}$) is also included for concise notations. It should be noted that values of $\mathbf{y}_n$ at $\mathcal{C}_n$ are not available during the sensing set selection stage, even if $\mathcal{C}_n$ is given. The problem is combinatorial in nature and it is NP-hard in general. In addition, the cost function is usually very complicated and it depends on the actual level set estimation algorithm.

The cost function in (4.4) can be alternatively expressed as

$$\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n-1}, \mathcal{C}_n] = \mathbb{E}_{\mathbf{y}_n}\left\{\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n}]\right\} \tag{4.5}$$

The reason that we take another layer of expectation with respect to $\mathbf{y}_n$ on the right hand side (RHS) of (4.5) is due to the fact that $\mathbf{y}_n$ is unknown before the selection of $\mathcal{C}_n$. The value of $\mathbf{y}_n$ will be revealed only after sensing samples are collected from $\mathcal{C}_n$ in time slot $t_n$.

With the alternative cost function expression in the RHS of (4.5), we can decompose the optimization in (4.4) into two steps. First, if $\mathcal{C}_n$ and $\mathbf{y}_n$ are known, identify the level set estimation algorithm that can minimize the the inner expectation on the RHS of (4.5), $\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n}]$. Second, select $\mathcal{C}_n$ that can minimize the overall cost function in (4.4). We will discuss the two steps in Sections 4.4 and 5.5, respectively.

## 4.4 Optimal Level Set Estimation with Known Measurement Results

In this section we present the optimal level set estimation algorithm under the condition that the measurements $\mathbf{y}^{1:n}$ collected from $\mathcal{C}^{1:n}$ are available at the FC. The algorithm will

be designed to minimize $\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n}]$ with the knowledge of $\mathbf{y}^{1:n}$, and the results will be used to facilitate the development of the active sensing algorithm in the next section.

Define $\mathbf{c}_{in} := [\mathbf{s}_i, t_n]$, $f_{in} := f([\mathbf{s}_i, t_n]^T)$, $\mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n}) := \mathbb{E}\left[f_{in}\left(\mathbf{f}^{1:n}\right)^T\right]$ and $\mathbf{K}(\mathcal{C}^{1:n}, \mathcal{C}^{1:n}) := \mathbb{E}\left[\mathbf{f}^{1:n}\left(\mathbf{f}^{1:n}\right)^T\right]$. The posteriori mean and covariance of $f_{in}$ given $\mathbf{y}^{1:n}$ is defined as

$$\hat{m}(\mathbf{c}_{in}) = \mathbb{E}\left[f(\mathbf{c}_{in})|\mathbf{y}^{1:n}\right] \tag{4.6}$$

$$\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}) = \mathbb{E}\left[f(\mathbf{c}_{in})f(\mathbf{c}_{jn})|\mathbf{y}^{1:n}\right] - \hat{m}(\mathbf{c}_{in})\hat{m}(\mathbf{c}_{jn}) \tag{4.7}$$

Due to the GP modeling, given $\mathcal{C}^{1:n}, \mathbf{y}^{1:n}$, the distribution of $f_{in}$ is still Gaussian, with the posteriori mean $\hat{m}(\mathbf{c}_{in})$ and variance $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ as [10]

$$\hat{m}(\mathbf{c}_{in}) = \mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n})[\mathbf{K}(\mathcal{C}^{1:n}, \mathcal{C}^{1:n}) + \sigma^2\mathbf{I}_{|\mathcal{C}^{1:n}|}]^{-1}\mathbf{Y}^{1:n} \tag{4.8}$$

$$\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) = k(\mathbf{c}_{in}, \mathbf{c}_{in}) - \mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n})[\mathbf{K}(\mathcal{C}^{1:n}, \mathcal{C}^{1:n}) + \sigma^2\mathbf{I}_{|\mathcal{C}^{1:n}|}]^{-1} \cdot \mathbf{K}(\mathcal{C}^{1:n}, \mathbf{c}_{in}) \tag{4.9}$$

where $\mathbf{I}_{|\mathcal{C}^{1:n}|}$ is an identity matrix of size $|\mathcal{C}^{1:n}|$.

The GP regression based level set estimation algorithm is given in Algorithm 3.

**Algorithm 3** GP regression based level set estimation

1: Input: $\mathcal{C}^{1:n}, \mathbf{y}^{1:n}$ at $t_n$.
2: Run GP regression for $\mathbf{c}_{in} = [\mathbf{s}_i, t_n], \forall \mathbf{s}_i \in \mathcal{X}$:

$$\hat{m}(\mathbf{c}_{in}) := \mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n})[\mathbf{K}(\mathcal{C}^{1:n}, \mathcal{C}^{1:n}) + \sigma^2\mathbf{I}_{|\mathcal{C}^{1:n}|}]^{-1}\mathbf{y}^{1:n}$$

3: Threshold $\hat{m}(\mathbf{c}_{in})$:

$$\hat{\mathcal{S}}_n = \{\mathbf{s}_i \in \mathcal{X} : \hat{m}(\mathbf{c}_{in}) > \gamma\}$$

4: Output $\hat{\mathcal{S}}_n$.

*Theorem 4.1*: The GP regression based level set estimation minimizes the expected estimation error with given $(\mathcal{C}^{1:n}, \mathbf{y}^{1:n})$, i.e., $\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathcal{C}^{1:n}, \mathbf{y}^{1:n}]$.

**Proof:** Since $\mathcal{S}_n$ is unknown for every $n$, the expression in (4.3) is not directly computable. Under the GP modeling on $\mathbf{f}, \mathbf{y}$, $\mathcal{S}_n$ is also a random process defined over $\mathcal{X}$. Given the observation history $(\mathcal{C}^{1:n}, \mathbf{y}^{1:n})$, we can always obtain the posterior distribution of $\mathcal{S}_n$. Therefore, the expected error with an estimation $\hat{S}$ at time $t_n$ is

$$\begin{aligned}\mathbb{E}[e(\hat{\mathcal{S}}_n)] &= \frac{1}{L}\sum_i \mathbb{P}\left[\mathbf{s}_i \in \Delta(\mathcal{S}_n, \hat{\mathcal{S}}_n)\right] \\ &= \frac{1}{L}\sum_i \Big(\mathbb{P}\left[f_{in} \le \gamma\right] \cdot \mathbb{I}\{\mathbf{s}_i \in \hat{\mathcal{S}}_n\} \\ &\qquad + \mathbb{P}\left[f_{in} > \gamma\right] \cdot \mathbb{I}\{\mathbf{s}_i \in \hat{\mathcal{S}}_n^c\}\Big) \end{aligned} \tag{4.10}$$

Therefore, the optimal estimator that minimizes (4.10) is to let

$$\mathbf{s}_i \in \begin{cases} \hat{\mathcal{S}} & \text{if } \mathbb{P}\left[f_{in} > \gamma\right] > \mathbb{P}\left[f_{in} \le \gamma\right] \\ \hat{\mathcal{S}}^c & \text{if } \mathbb{P}\left[f_{in} > \gamma\right] \le \mathbb{P}\left[f_{in} \le \gamma\right] \end{cases} \tag{4.11}$$

for every $\mathbf{s}_i \in \mathcal{X}$.

Since $f_{in}$ given $\mathbf{y}^{1:n}$ is Gaussian distributed, we have

$$\mathbb{P}\left[f_{in} > \gamma|\mathbf{y}^{1:n}\right] = Q\left(\frac{\gamma - \hat{m}(\mathbf{c}_{in})}{\sqrt{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}\right) \tag{4.12}$$

where $Q(\cdot)$ is the Gaussian $Q$ function, and the posteriori mean $\hat{m}(\mathbf{c}_{in})$ and variance $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ are given in (4.8) and (4.9), respectively. The optimal estimator defined in (4.11) is then

reduced to compare $\hat{m}(\mathbf{c}_{in})$ with $\gamma$. If $\hat{m}(\mathbf{c}_{in}) > \gamma$, the probability in (4.12) is greater than 0.5, thus, we should let $\mathbf{s}_i \in \hat{\mathcal{S}}_n$; otherwise, we let $\mathbf{s}_i \in \hat{\mathcal{S}}_n^c$. ■

*Corollary 4.1*: The expected error given by Algorithm 3 with given $(\mathcal{C}^{1:n}, \mathbf{y}^{1:n})$ is

$$\mathbb{E}[e(\hat{\mathcal{S}}, t_n)|\mathbf{y}^{1:n}] = \frac{1}{L}\sum_i Q\left(\frac{|\gamma - \hat{m}(\mathbf{c}_{in})|}{\sqrt{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}\right) \tag{4.13}$$

Eqns. (4.8) and (4.9) capture the relationship between the samples $\mathbf{y}^{1:n}$, and the underlying function value $f_{in}$. The posterior distribution of $f_{in}$ depends on the correlation between $f_{in}$ and $\mathbf{f}^{1:n}$ (through $\mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n})$), the correlation between the sensing samples (through $\mathbf{K}(\mathcal{C}^{1:n}, \mathcal{C}^{1:n})$), and the noise level in sensing and transmission process. We note that $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ depends on $\mathcal{C}^{1:n}$ only and $\hat{m}(\mathbf{c}_{in})$ depends on both $\mathcal{C}^{1:n}$ and $\mathbf{y}^{1:n}$.

The optimality of Algorithm 3 is conditioned upon the fact that $\mathcal{C}_n$ and $\mathbf{y}_n$ are given. We will discuss how to actively select $\mathcal{C}_n$ based on sensing history $(\mathcal{C}^{1:n-1}, \mathbf{y}^{1:n-1})$ in the next section.

## 4.5 Optimal Active Sensing for Level Set Estimation

In this section, we consider a scenario where the FC is able to coordinate with the sensor nodes and actively selects the sensing locations in each time slot, such that the cost function in (4.4) is minimized.

The active sensing scheme consists of two steps in each time slot $t_n$. The first step is to obtain an initial estimate of the distribution of $f(\mathbf{c}_{in})$, i.e., its mean $\bar{m}(\mathbf{c}_{in})$, and covariance

$\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{jn})$, based on the sensing history up to to $t_{n-1}$, i.e., $(\mathcal{C}^{1:n-1}, \mathbf{y}^{1:n-1})$, that is

$$\bar{m}(\mathbf{c}_{in}) = \mathbb{E}\left[f_{in}|\mathbf{y}^{1:n-1}\right] \tag{4.14}$$

$$\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}) = \mathbb{E}\left[f_{in}f_{jn}|\mathbf{y}^{1:n-1}\right] - \bar{m}(\mathbf{c}_{in})\bar{m}(\mathbf{c}_{jn}) \tag{4.15}$$

It should be noted that $\hat{m}(\mathbf{c}_{in})$ and $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{jn})$ defined in (4.8) and (4.9) are conditioned upon $\mathbf{y}^{1:n}$, and they are different from $\bar{m}(\mathbf{c}_{in})$ and $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{jn})$ defined in (4.14) and (4.15), which are conditioned upon $\mathbf{y}^{1:n-1}$.

The observation history up to time slot $t_{n-1}$ is thus utilized to provide a rough sketch of the level set. Then, the second step is to sample the sensing field $\mathcal{X}$ in $t_n$, i.e., to select a subset of up to $N$ locations to make observations, and refine the level set estimate based on Algorithm 3 once samples are collected from $\mathcal{C}_n$. The objective of the two-step active sensing is to minimize the *expected* estimation error averaged over $\mathbf{y}_n$, under a cardinality constraint on $\mathcal{C}_n$. Details of the two steps are provided in the following subsections.

The major difficulty of the active sensing scheme lies in the step of selecting $\mathcal{C}_n$. In order to evaluate the impact of the selection of $\mathcal{C}_n$ on the final expected level set estimation error, i.e., the objective function in (4.4), we first decompose the estimation error in (4.13) as a function of $(\mathcal{C}_n, \mathbf{y}_n)$ and $(\mathcal{C}^{1:n-1}, \mathbf{y}^{1:n-1})$.

Based on the GP assumption, the posteriori mean $\bar{m}(\mathbf{c}_{in})$ and variance $\bar{k}(\mathbf{c}_{in}, \mathbf{x}_{jn})$ defined

in (4.14) and (4.15) can be written as

$$\bar{m}(\mathbf{c}_{in}) = \mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n-1})[\mathbf{K}(\mathcal{C}^{1:n-1}, \mathcal{C}^{1:n-1}) + \sigma^2 \mathbf{I}_{|\mathcal{C}^{1:n-1}|}]^{-1} \mathbf{Y}^{1:n-1} \tag{4.16}$$

$$\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}) = k(\mathbf{c}_{in}, \mathbf{c}_{jn}) - \mathbf{K}(\mathbf{c}_{in}, \mathcal{C}^{1:n-1})[\mathbf{K}(\mathcal{C}^{1:n-1}, \mathcal{C}^{1:n-1}) + \sigma^2 \mathbf{I}_{|\mathcal{C}^{1:n-1}|}]^{-1} \mathbf{K}(\mathcal{C}^{1:n-1}, \mathbf{c}_{jn}). \tag{4.17}$$

The difference between $(\bar{m}(\mathbf{c}_{in}), \bar{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}))$ and $(\hat{m}(\mathbf{c}_{in}), \hat{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}))$ is directly related to the impacts of selecting $\mathcal{C}_n$ on the expected level set estimation error. To identify the relationship, we define the following variables.

$$h(\mathbf{c}_{in}, \mathcal{C}_n) := \bar{\mathbf{K}}(\mathbf{c}_{in}, \mathcal{C}_n)[\bar{\mathbf{K}}(\mathcal{C}_n, \mathcal{C}_n) + \sigma^2 \mathbf{I}_{|\mathcal{C}_n|}]^{-1}[\mathbf{y}_n - \bar{\mathbf{m}}(\mathcal{C}_n)] \tag{4.18}$$

$$\sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n) := \bar{\mathbf{K}}(\mathbf{c}_{in}, \mathcal{C}_n)[\bar{\mathbf{K}}(\mathcal{C}_n, \mathcal{C}_n) + \sigma^2 \mathbf{I}_{|\mathcal{C}_n|}]^{-1} \bar{\mathbf{K}}(\mathcal{C}_n, \mathbf{c}_{in}) \tag{4.19}$$

where the elements of the posterior mean vector $\bar{\mathbf{m}}(\mathcal{C}_n) = \mathbb{E}(\mathbf{f}_n | \mathbf{y}^{1:n-1})$ is defined in (4.16), and

$$\bar{\mathbf{K}}(\mathbf{c}_{in}, \mathcal{C}_n) = \mathbb{E}\left\{[f_{in} - \bar{m}(\mathbf{c}_{in})][\mathbf{f}_n - \bar{\mathbf{m}}(\mathcal{C}_n)]^T | \mathbf{y}^{1:n-1}\right\}$$

$$\bar{\mathbf{K}}(\mathcal{C}_n, \mathcal{C}_n) = \mathbb{E}\left\{[\mathbf{f}_n - \bar{\mathbf{m}}(\mathcal{C}_n)][\mathbf{f}_n - \bar{\mathbf{m}}(\mathcal{C}_n)]^T | \mathbf{y}^{1:n-1}\right\}$$

are the posterior covariance vector and matrix with elements $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{jn})$ defined in (4.17).

With the notation in (4.18) and we have the following theorem regarding the decomposition of $\hat{m}(\mathbf{c}_{in})$ and $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$.

*Theorem 4.2*: The posteriori mean and variance, $\hat{m}(\mathbf{c}_{in})$ and $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$, defined in (4.8) and (4.9) can be decomposed in the following form

$$\hat{m}(\mathbf{c}_{in}) = \bar{m}(\mathbf{c}_{in}) + h(\mathbf{c}_{in}, \mathcal{C}_n) \tag{4.20}$$

$$\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) = \bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) - \sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n) \tag{4.21}$$

**Proof:** The proof relies on the conditional distribution of jointly Gaussian distributed random variables. Consider three jointly Gaussian distributed random vectors, $\mathbf{x}$, $\mathbf{z}_1$, and $\mathbf{z}_2$, and we have the following relationship

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1} + \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}_2|\mathbf{z}_1}\boldsymbol{\Sigma}^{-1}_{\mathbf{z}_2\mathbf{z}_2|\mathbf{z}_1}\left(\mathbf{z}_2 - \boldsymbol{\mu}_{\mathbf{z}_2|\mathbf{z}_1}\right), \tag{4.22}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1} - \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}_2|\mathbf{z}_1}\boldsymbol{\Sigma}^{-1}_{\mathbf{z}_2\mathbf{z}_2|\mathbf{z}_1}\boldsymbol{\Sigma}_{\mathbf{z}_2\mathbf{x}|\mathbf{z}_1}, \tag{4.23}$$

where the notations, $\boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}} = \mathbb{E}(\mathbf{a}|\mathbf{b})$ and $\Sigma_{\mathbf{a}\mathbf{b}|\mathbf{c}} = \mathbb{E}[(\mathbf{a} - \boldsymbol{\mu}_{\mathbf{a}|\mathbf{c}})(\mathbf{b} - \boldsymbol{\mu}_{\mathbf{b}|\mathbf{c}})^T|\mathbf{c}]$, are used in the above expressions, with $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ being three random vectors.

If we let $\mathbf{x} = f_{in}$, $\mathbf{z}_1 = \mathbf{y}^{1:n-1}$, and $\mathbf{z}_2 = \mathbf{y}_n = \mathbf{f}_n + \boldsymbol{\xi}$, then

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \hat{m}(\mathbf{c}_{in}), \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1\mathbf{z}_2} = \hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) \tag{4.24}$$

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}_1} = \bar{m}(\mathbf{c}_{in}), \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|\mathbf{z}_1} = \bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in}). \tag{4.25}$$

In addition, it can be easily shown that $\mu_{\mathbf{z}_2|\mathbf{z}_1} = \bar{\mathbf{m}}(\mathcal{C}_n)$, and

$$\mathbf{\Sigma}_{\mathbf{x}\mathbf{z}_2|\mathbf{z}_1} = \bar{\mathbf{K}}(\mathbf{c}_{in}, \mathcal{C}_n) \tag{4.26}$$

$$\mathbf{\Sigma}_{\mathbf{z}_2\mathbf{z}_2|\mathbf{z}_1} = \bar{\mathbf{K}}(\mathcal{C}_n, \mathcal{C}_n) + \sigma^2 \mathbf{I}_{|\mathcal{C}_n|} \tag{4.27}$$

Substituting the above equations into (4.22) and (4.23) yields (4.20) and (4.21). ■

We note that $\bar{m}(\mathbf{c}_{in})$ and $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ depend on $(\mathcal{C}^{1:n-1}, \mathbf{y}^{1:n-1})$ only, $\sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n)$ depends on $\mathcal{C}^{1:n-1}$ and $\mathcal{C}_n$, while $h(\mathbf{c}_{in}, \mathcal{C}_n)$ depends on $(\mathcal{C}^{1:n-1}, \mathbf{y}^{1:n-1})$ and $(\mathcal{C}_n, \mathbf{y}_n)$. Moreover, given $(\mathcal{C}^{1:n-1}, \mathbf{y}^{1:n-1})$ and $(\mathcal{C}_n, \mathbf{y}_n)$, $h(\mathbf{c}_{in}, \mathcal{C}_n)$ is a Gaussian random variable with zero mean and variance $\sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n)$. As a result, the results in Theorem 4.2 decompose $\hat{m}(\mathbf{c}_{in})$ and $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ into two parts, one part depends on $\mathcal{C}^{1:n-1}$, and the other part depends on $\mathcal{C}_n$.

In order to simplify notations, we define

$$\gamma_i := \frac{\bar{m}(\mathbf{c}_{in}) - \gamma}{\sqrt{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}, \quad \sigma_i := \frac{\sigma_h(\mathbf{c}_{in}, \mathcal{C}_n)}{\sqrt{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}} \tag{4.28}$$

With the decomposition given in Theorem 4.2, we can establish an explicit relationship between the cost function in (4.4) of the optimization problem and the optimization variable $\mathcal{C}_n$ as shown in the following theorem. The explicit expression of the cost function enables us to solve the optimization problem.

*Theorem 4.3*: The cost function in (4.4) with respect to the sensing location $\mathcal{C}_n$ is

$$\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n-1}, \mathcal{C}_n] = \frac{1}{\pi L} \sum_{i=1}^{L} \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 + \frac{\sigma_i^2}{(1-\sigma_i^2)\sin^2(\theta)}}} \times \exp\left\{ -\frac{\gamma_i^2}{2[\sigma_i^2 + (1-\sigma_i^2)\sin^2(\theta)]} \right\} d\theta \tag{4.29}$$

where $\sigma_i$ depends on $\mathbf{y}^{1:n-1}$ and $\mathcal{C}_n$, and it is defined in (4.28).

**Proof:** From (4.5), Corollary 4.1, and Theorem 4.2, the cost function in (4.4) can be alternatively expressed as

$$\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n-1}, \mathcal{C}_n] = \mathbb{E}_Y[Q(|Y|)] \tag{4.30}$$

where

$$Y = \frac{h(\mathbf{c}_{in}, \mathcal{C}_n) + \bar{m}(\mathbf{c}_{in}) - \gamma}{\sqrt{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) - \sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n)}}. \tag{4.31}$$

Since $h(\mathbf{c}_{in}, \mathcal{C}_n) \sim \mathcal{N}(0, \sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n))$, it is straightforward that the random variable $Y$ is Gaussian distributed with mean $\mu_Y = \frac{\gamma_i}{\sqrt{1-\sigma_i^2}}$ and variance $\sigma_Y^2 = \frac{\sigma_i^2}{1-\sigma_i^2}$.

With Craig's formula of the Gaussian $Q$-function [6], we have

$$\mathbb{E}_Y[Q(|Y|)] = \frac{1}{\pi} \int_0^{\frac{\pi}{2}} \mathbb{E}_Y \left[ e^{-\frac{Y^2}{2\sin^2(\theta)}} \right] d\theta \tag{4.32}$$

Since $Y$ is Gaussian distributed, $Y^2/\sigma_Y^2$ is non-central $\chi^2$-distributed with one degree-of-freedom. The moment generating function (MGF) of $Y^2/\sigma_Y^2$ is

$$M(t) = \mathbb{E}\left[e^{\frac{tY^2}{\sigma^2}}\right] = \exp\left\{\frac{\mu_Y^2 t}{(1-2t)\sigma_Y^2}\right\}\frac{1}{\sqrt{1-2t}} \tag{4.33}$$

Combining (4.32) with (4.33) yields

$$\mathbb{E}_Y[Q(|Y|)]=\frac{1}{\pi}\int_0^{\frac{\pi}{2}} \exp\left\{-\frac{\mu_Y^2}{2(\sigma_Y^2+\sin^2(\theta))}\right\}\frac{1}{\sqrt{1+\frac{\sigma_Y^2}{\sin^2(\theta)}}}d\theta$$

This completes the proof. ■

Theorem 4.3 gives the exact explicit expression of the cost function of the optimization problem. Even though it is expressed as an integration, the integral is of finite limits and the integrand contains only elementary functions, thus it can be easily evaluated numerically. The cost function in Theorem 4.3 is expressed as a function of $\gamma_i$ and $\sigma_i$ defined in (4.28). It should be noted that $\gamma_i$ is independent of the choice of $\mathcal{C}_n$. So the sensing set selection will only affect the cost function through $\sigma_i$.

Based on the result in Theorem 4.3, the optimization problem in (4.4) can be solved by using an exhaustive search algorithm as shown in Algorithm 4.

**Algorithm 4** An exhaustive search algorithm

1: Input: $\mathcal{C}^{1:n-1}$, $\mathbf{y}^{1:n-1}$, $\mathcal{X}$ with size $L$.
2: Calculate $\bar{\mathbf{m}}$ with (4.16) and $\bar{\mathbf{K}}$ with (4.17).
3: Calculate $\gamma_i$ with (4.28), for $i = 1, \cdots, L$.
4: **for** each possible combination of $\mathcal{C}_n \in \mathcal{X}$ **do**
5: Calculate $\sigma_i$ with (4.28), for $i = 1, \cdots, L$
6: Calculate the cost function with (4.29).
7: **end for**
8: Output $\mathcal{C}_n$ that minimizes the cost function.

The Algorithm exhaustively searches all the possible $\binom{L}{N}$ sets of $\mathcal{C}_n$, and finds the one that minimizes the cost function. The complexity becomes prohibitive as $L$ and/or $N$ becomes large.

## 4.6 Greedy Algorithms for Active Sensing

Greedy algorithms for active sensing are presented in this section to achieve a balanced tradeoff between performance and complexity.

### 4.6.1 A Cost Function Upper Bound

The greedy algorithms are developed by using an upper bound of the cost function in (4.29). The upper bound is obtained by applying $\sin\theta \leq 1$ to (4.29), and the result is as follows.

*Corollary 4.2*: The cost function in (4.4) with respect to the sensing location $\mathcal{C}_n$ is upper bounded by

$$\mathbb{E}[e(\hat{\mathcal{S}}_n)|\mathbf{y}^{1:n-1}, \mathcal{C}_n] \leq \frac{1}{2L}\sum_{i=1}^{L} \exp\left\{-\frac{\gamma_i^2}{2}\right\} \cdot \tau_i \tag{4.34}$$

where $\tau_i = \sqrt{1-\sigma_i^2} = \sqrt{1 - \frac{\sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n)}{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}$.

In the upper bound (4.34), the variables $\gamma_i$ and $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ are independent of the choice of $\mathcal{C}_n$. Only $\sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n)$ depends on $\mathcal{C}_n$. Compared with (4.29), the upper bound (4.34) is a linear function in $\tau_i$, thus much easier to evaluate.

Therefore, we propose to minimize the upper bound instead. Define $\alpha_i = \frac{1}{2}\exp\left\{-\frac{\gamma_i^2}{2}\right\}$.

Then, a new optimization problem with the upper bound can be formulated as

$$\begin{aligned} \min_{\mathcal{C}_n} \quad & \frac{1}{L}\sum_i \alpha_i \sqrt{1-\frac{\sigma_h^2(\mathbf{c}_{in},\mathcal{C}_n)}{\bar{k}(\mathbf{c}_{in},\mathbf{c}_{in})}} \\ \text{s.t.} \quad & |\mathcal{C}_n| \le N \end{aligned} \tag{4.35}$$

The objective function in (4.35) coincides with our intuition. The weight $\alpha_i$ is a function of $\gamma_i$, which is defined in (4.28). We note that in its definition, the numerator $|\bar{m}(\mathbf{c}_{in}) - \gamma|$ measures the deviation of $\bar{m}(\mathbf{c}_{in})$ from the threshold $\gamma$, which is then normalized by $\sqrt{\bar{k}(\mathbf{c}_{in},\mathbf{c}_{in})}$, the estimated standard deviation. The larger the value of $\gamma_i$, the less likely an classification error will happen at $\mathbf{s}_i$. This is reflected by $\alpha_i$, since it decreases in $\gamma_i$. With a small weight $\alpha_i$, the term $\alpha_i\tau_i$ plays a less important role in the optimization (4.35). The solution to (4.35) thus automatically allocates more resources to the locations with heavy weights $\alpha_i$s.

At the beginning of each time slot, the system obtains an initial estimate of $f(\mathbf{x})$, characterized as $(\bar{\mathbf{m}}, \bar{\mathbf{K}})$. Intuitively, if the initially estimated mean $\bar{m}(\mathbf{c}_{in})$ deviates significantly relative to variance $\bar{k}(\mathbf{c}_{in},\mathbf{c}_{in})$ from the threshold $\gamma$, the probability of incorrectly classifying $\mathbf{s}_i$ in slot $t_n$ is very small, and bringing in another sample from $\mathbf{s}_i$ will not help much in terms of the expected error; on the other hand, if $\bar{m}(\mathbf{c}_{in})$ is quite close to the threshold $\gamma$, sensing around $\mathbf{s}_i$ potentially can make the classification much more accurate. Therefore, minimizing the level set estimation error is not equivalent to minimizing the total posterior variance. Essentially, to estimate the level set of a function is to search for the *boundary* of the level sets. For sensing locations far away from the boundary, their actual values do not have much impact on the level set estimation, thus more sensing resources should be allocated for

locations around the boundary. Such an approach has the potential to significantly reduce the number of required sensing samples because the area of boundary is usually only a very small percentage of the total area, thus achieving sparse sampling.

### 4.6.2 Greedy Algorithms

Even though the optimization problem (4.35) has a much less complicated form than (4.4), it is still an NP-hard problem. In the following, we propose Algorithm 5 to solve it in a greedy fashion.

**Algorithm 5** A greedy algorithm

Input: $\mathcal{C}^{1:n-1}$, $\mathbf{y}^{1:n-1}$, $\mathcal{C}_n = \emptyset$, $\mathcal{F} = \mathcal{X}$.
Calculate $\bar{\mathbf{m}}$ with (4.16) and $\bar{\mathbf{K}}$ with (4.17).
$\hat{\mathbf{K}} \leftarrow \bar{\mathbf{K}}$.
Calculate $\gamma_i$ and $\alpha_i$, for $i = 1, \cdots, L$.
**for** $k = 1, 2, \ldots, N$ **do**
    Calculate $\sigma_h^2(\mathbf{c}_{in}, \mathbf{c}_{jn}) = \frac{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{jn})^2}{\hat{k}(\mathbf{c}_{jn}, \mathbf{c}_{jn}) + \sigma^2}$ for $\mathbf{c}_{jn} \in (\mathcal{F}, t_n)$

$$l = \arg\min_j \frac{1}{L} \sum_i \frac{\alpha_i \sqrt{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) - \sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_j)}}{\sqrt{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}} \tag{4.36}$$

    $\mathcal{C}_n \leftarrow \mathcal{C}_n \cup l$, $\mathcal{F} \leftarrow \mathcal{F} \backslash l$.
    Update $\hat{\mathbf{K}}$: for $i, j = 1, 2, \ldots, L$,

$$\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}) \leftarrow \hat{k}(\mathbf{c}_{in}, \mathbf{c}_{jn}) - \frac{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{ln}) \hat{k}(\mathbf{c}_{ln}, \mathbf{c}_{jn})}{\hat{k}(\mathbf{c}_{ln}, \mathbf{c}_{ln}) + \sigma^2}$$

**end for**
Output $\mathcal{C}_n$.

In this algorithm, the optimization problem is solved in a sequential and greedy fashion. Specifically, we select one sensing location from $\mathcal{X}$ in each iteration according to (4.36), with the objective to minimize the objective function in (4.35). We point out that the objective function in (4.36) is different from that in (4.35), as in each iteration, we need to remove the

impacts from sensing locations already included in $\mathcal{C}_n$. Thus, we use $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ instead of $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ in the numerator in (4.36). However, we keep $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ in the denominator fixed during the selection of $\mathcal{C}_n$ to ensure each term is normalized by the same factor as in (4.35). The impact of a sensing location on the posterior variance on every $\mathbf{s}_i$ can be explicitly evaluated through $\sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_j)$. Once one sensing location is selected, the posterior covariance matrix $\hat{\mathbf{K}}$ is updated to remove the impact from the newly added sensing location. After that, another iteration is performed with the updated covariance matrix.

We note that individual terms in the summation in (4.35) are coupled due to the cross correlation between $\mathbf{y}_n$ and $f(\mathbf{x})$ carried through $\sigma_h^2(\mathbf{c}_{in}, \mathcal{C}_n)$. Therefore, a sample collected from $\mathbf{s}_i$ does not only *directly* affect the estimation accuracy at $\mathbf{s}_i$, but also *indirectly* affects locations nearby. The optimization requires us to jointly consider the *direct* and *indirect* impacts of all of the samples, which makes the problem complicated.

To simplify the optimization, we ignore the *indirect* impacts of samples, and focus on direct impacts only. This is equivalent to neglecting the cross correlations between different locations. This results in a simplified greedy algorithm presented in Algorithm 6.

Specifically, to select the first sensing location, we mask out the off-diagonal entries in $\bar{\mathbf{K}}$, and this leads to the following approximation

$$\sigma_h^2(\mathbf{c}_{in}, \mathbf{c}_{jn}) = \frac{\bar{k}^2(\mathbf{c}_{in}, \mathbf{c}_{jn})}{\bar{k}(\mathbf{c}_{jn}, \mathbf{c}_{jn}) + \sigma^2}\delta_{ij} \tag{4.38}$$

**Algorithm 6** A simplified greedy algorithm

Input: $\mathcal{C}^{1:n-1}$, $\mathbf{y}^{1:n-1}$, $\mathcal{C}_n = \emptyset$, $\mathcal{F} = \mathcal{X}$.
Update the distribution of $f(\mathbf{c})$, obtain $\bar{\mathbf{m}}, \bar{\mathbf{K}}$.
Assign $\bar{\mathbf{K}}$ to $\hat{\mathbf{K}}$.
Calculate $\gamma_i$, $\alpha_i$, for $i = 1, \cdots, L$.
**for** $k = 1, 2, \ldots, N$ **do**
    Calculate $\sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_i) = \frac{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})^2}{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) + \sigma^2}$ for $\mathbf{s}_i \in \mathcal{F}$

$$l = \arg\max_{i \in \mathcal{F}} \frac{\alpha_i \sqrt{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}{\sqrt{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}} \left(1 - \sqrt{1 - \frac{\sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_i)}{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}\right) \tag{4.37}$$

    $\mathcal{C}_n \leftarrow \mathcal{C}_n \cup l$, $\mathcal{F} \leftarrow \mathcal{F} \backslash l$.
    Update $\bar{\mathbf{K}}$: for $i, j = 1, 2, \ldots, L$,

$$\hat{k}(\mathbf{c}_{in}, \mathbf{x}_{jn}) \leftarrow \hat{k}(\mathbf{c}_{in}, \mathbf{x}_{jn}) - \frac{\hat{k}(\mathbf{c}_{in}, \mathbf{x}_{ln}) \hat{k}(\mathbf{x}_{ln}, \mathbf{x}_{jn})}{\hat{k}(\mathbf{x}_{ln}, \mathbf{x}_{ln}) + \sigma^2}$$

**end for**
Output $\mathcal{C}_n$.

Thus, the single sensing location that minimizes (4.35) must satisfy

$$\begin{aligned} l &= \arg\min_i \sum_{j \neq i} \alpha_j + \alpha_i \sqrt{1 - \frac{\sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_i)}{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}} \\ &= \arg\max_i \alpha_i \left(1 - \sqrt{1 - \frac{\sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_i)}{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}\right) \end{aligned} \tag{4.39}$$

Once this location is selected, we then take its *indirect* impact on the other nodes into consideration by updating the posterior covariance matrix $\bar{\mathbf{K}}$ conditional on $l$. We point out that in (4.37) we use $\sqrt{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})} - \sqrt{\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in}) - \sigma_h^2(\mathbf{c}_{in}, \mathbf{s}_i)}$ to measure the performance *gain* if a sample is collected at $\mathbf{s}_i$, which is then weighted by $\frac{\alpha_i}{\sqrt{\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})}}$. We use $\hat{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ instead of $\bar{k}(\mathbf{c}_{in}, \mathbf{c}_{in})$ in the *gain* to remove the *indirect* impacts from sensing locations already included in $\mathcal{C}_n$.

Due to the fast decaying spatial correlation, we expect that the *indirect* gain brought by a sample is localized and roughly proportionally to its *direct* gain, which makes (4.38) a valid approximation.

## 4.7 Simulation and Experiment Results

In this section, we evaluate the performance of the proposed active sensing algorithms through simulations and experiments, and compare them with a baseline passive sensing algorithm and a batch sample selection level set estimation algorithm [9] based on the Gaussian Process Upper Confidence Bound (GP-UCB) method [30]. In the passive sensing algorithm, the FC does not coordinate with sensor nodes for sensing. Rather, in each time slot, sensor nodes from $N$ random locations sense the field and send measurements to the FC. The FC then performs level set estimation according to Algorithm 3. In the GP-UCB level set estimation algorithm, the $N$ sensor node locations are selected according to the batch sample selection proposed in [9], which minimizes a cost function that penalizes both the posterior variance and the deviation from the level set boundary.

### 4.7.1 Simulation Results

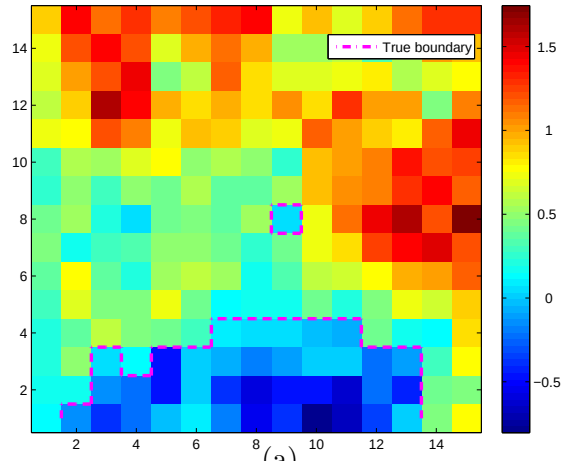
We consider a sensor network in a 2-D squared area. The sensing field is partitioned to $L = d \times d$ segments with unit length each. The covariance function is selected as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \rho_s^{\|\mathbf{s}_i - \mathbf{s}_j\|} \cdot \rho_t^{|t_i - t_j|}, \tag{4.40}$$

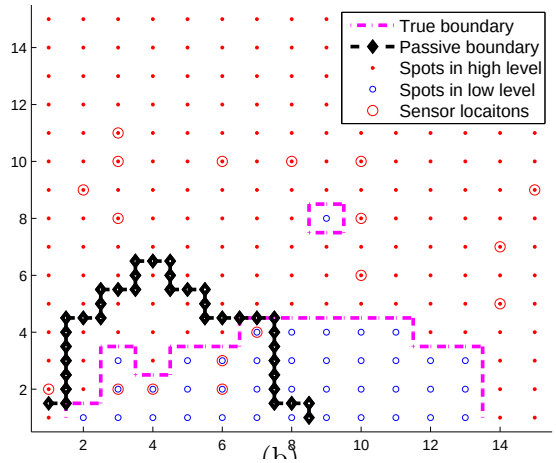
where $\rho_s$ and $\rho_t \in [0, 1]$ are the spatial and temporal correlation coefficients, respectively. The signal to sensing noise ratio is 30 dB, and the level set threshold is $\gamma = 0.1$.

#### 4.7.1.1 Comparison on different sensing location selections

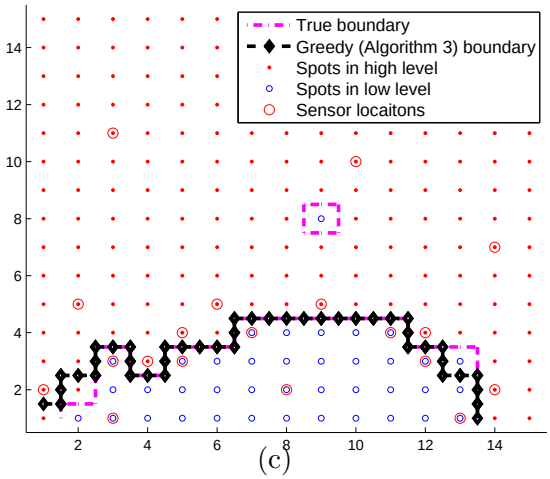
To illustrate the difference in sensing decision making between our proposed algorithms and the passive sensing algorithm, we first consider a special scenario, where $\rho_t = 1$, $L = 225$ ($d = 15$) and $N = 1$. This may correspond to a temporally slow-varying sensing field and the time interval between any two consecutive sensing actions is small and thus negligible. We set $\rho_s = 0.96$. The heatmap of the original signal is shown in Fig. 4.1a. We reconstruct the original signal based on the first 20 locations selected by the passive sensing algorithm in Fig. 4.1b, and those selected by Algorithm 5 in Fig. 4.1c. The boundaries of the reconstructed level sets are plotted, and they are compared to the underlying ground truth in Figs. 4.1b and 4.1c.




(a)




(b)




(c)

Figure 4.1: (a) Heatmap of the original signal. (b) and (c) Red circles represent 20 sensing locations selected by the sensing algorithms, in a filed partitioned into $15 \times 15 = 225$ segments. The spots in the original truth signal are represented as red solid dots in the high level sets and blue circles in the low level sets.

As illustrated in Fig. 4.1b, the random sensing location selection in the passive sensing

scheme renders a relatively uniform distribution of sensing locations. This results in a good reconstruction of the function. The MSE of the function reconstruction is 0.0924. On the other hand, in Fig. 4.1c, most of the sensing locations of the greedy algorithm are around the boundary. As a result, the greedy algorithm gets a more accurate estimation of the boundary. This matches with our optimization objective, as the accurate identification of the boundary, rather than accurate function reconstruction, plays a critical role for level set estimation. Even though the MSE for the signal reconstructed in the greedy algorithm (MSE = 0.0969) is higher than the passive approach, it has a much lower level set estimation error. In this example, the average level set estimation error from passive sensing and the greedy Algorithm 3 is 0.1556 and 0.0178, respectively. We also applied the GP-UCB algorithm [9,30] in the simulations. The MSE of the GP-UCB algorithm is 0.0813, the smallest among the three. The level set estimation error of GP-UCB is 0.0467, better than the passive algorithm, but worse than our newly proposed greedy Algorithm 3. All MSE and level set estimation error results here are calculated by using the results in Figs. 4.1b and 4.1c in one trial.

For further comparisons, Table I shows the mean, the first, second, and third quantiles of LSE error and MSE obtained by using 1,000 Monte-Carlo trials. The system configurations are the same as in Fig. 1. The results show that the greedy algorithm has the best performance in terms of level set estimation error, followed by the GP-UCB algorithm and the passive algorithm. On the other hand, the greedy algorithm has the worst performance in terms of MSE.

Table I. Mean and quantiles of LSE error and MSE for 1,000 Monte-Carlo trials

| | | Mean | Q1 | Q2 | Q3 |
|---|---|---|---|---|---|
| LSE error | Passive | 0.1465 | 0.0844 | 0.1422 | 0.2000 |
| | GP-UCB | 0.1293 | 0.0711 | 0.1244 | 0.1822 |
| | Algorithm 3 | 0.1124 | 0.0578 | 0.1067 | 0.1578 |
| MSE | Passive | 0.1975 | 0.1625 | 0.1880 | 0.2194 |
| | GP-UCB | 0.1806 | 0.1532 | 0.1756 | 0.2028 |
| | Algorithm 3 | 0.2209 | 0.1779 | 0.2095 | 0.2487 |

#### 4.7.1.2 Robustness of Gaussian Assumption

The algorithms are developed by assuming that the data can be modeled by a multivariate Gaussian process. To verify the robustness of the Gaussian assumption, we apply the algorithms to a group of data generated by following the Laplace distribution. Except the distributions, all other settings are the same as in Fig. 4.1. That is, the data following the Laplace distribution have the same mean and covariance matrix as those following the Gaussian distribution. The average LSE errors are obtained by averaging over 1,000 Monte-Carlo simulations, and the results are presented in Table II. The average LSE errors for data generated by following the Laplace distribution are slightly higher than those following the Gaussian distribution. Compared to data with Gaussian distribution, the average LSE errors for data with the Laplace distribution increase by 11.9%, 11.9%, and 14.4% for the passive sensing, GP-UCB, and the greedy algorithm (Algorithm 3), respectively. These results show that even though the proposed algorithms are developed based on the Gaussian assumption, they can still be applied to data following non-Gaussian distributions, at the cost of a slight increase in the LSE error probability.

Table II. Mean of LSE Error for data following Gaussian and Laplace distributions

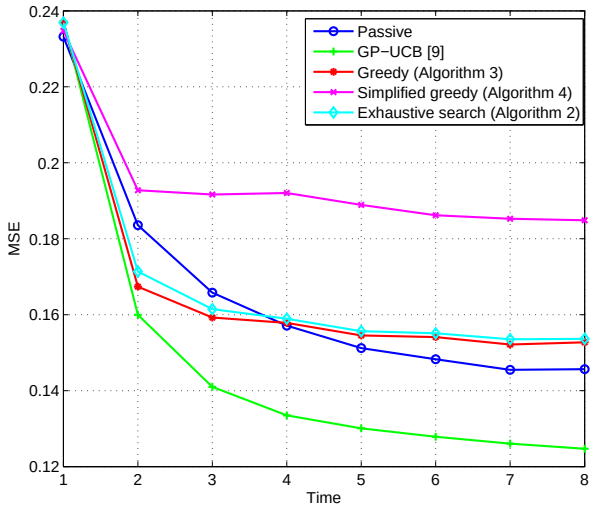
| | Gaussian distribution | Laplace distribution |
|---|---|---|
| Passive | 0.1465 | 0.1639 |
| GP-UCB | 0.1293 | 0.1448 |
| Algorithm 3 | 0.1124 | 0.1286 |

#### 4.7.1.3 Effect of time window size on the performance

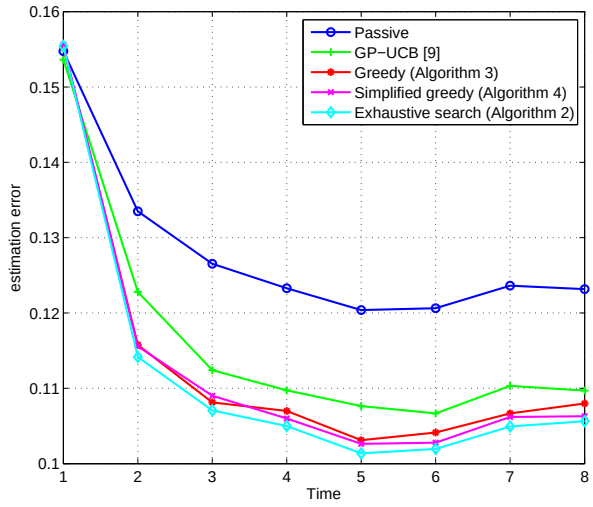
As time progresses, more and more samples are collected and added to (4.8) for estimation. This quickly becomes formidable due to the high storage requirement and computational complexity. One the other hand, under the assumption that the temporal correlation decays exponentially in $t$, samples collected in the past have less and less impacts on the estimation as time progresses. This motivates us to adopt a truncated version of the level set estimation algorithm described in Algorithm 3. Specifically, we propose a sliding window scheme, which only keeps samples collected in the most recent $T$ time slots for the regression in Algorithm 3.

In order to explicitly evaluate the effect of window size on the sensing and inference performance, we perform the following simulation. We set $\rho_t = 0.9$, $\rho_s = 0.9$, $L = 169$ ($d = 13$) and fix the total number of samples collected in each slot as $N = 15$. The window size $T$ varies from 1 to 8. When $T = 1$, it means that only the current sensing samples are used for level set estimation. Thus the active sensing algorithms including GP-UCB algorithm become identical to the passive sensing algorithm in this case. When $T > 1$, samples collected from the most recent $T$ slots are used in the regression. For each $T$, we randomly generate 1,000 different traces, and obtain the average results. The reconstruction MSE as a function of $T$ is plotted in Fig. 4.2a, and the average level set estimation error is plotted in Fig. 4.2b.

In Fig. 4.2a, when $T \geq 4$, the GP-UCB algorithm has the best MSE performance, followed by the passive algorithm, the greedy algorithm (Algorithm 5), the exhaustive search algorithm that employs exhaustive search (Algorithm 4), and the simplified greedy algorithm (Algorithm 6). On the other hand, in Fig. 4.2b for level set estimation errors, the proposed Algorithms 4 – 6 achieve significant performance gains over the passive and the GP-UCB algorithms. The objective of the proposed algorithms is to estimate the level set rather than to reconstruct the underlying function. The information provided from previous samples guides the sample selection decision, which may even worsen the MSE performance. In addition, the performance of the greedy algorithm (Algorithm 5) is very close to its simplified version (Algorithm 6), and both can nearly achieve the optimum performance of the exhaustive search algorithm. For all three algorithms and for this particular choice of $\rho_t = 0.9$, the MSE and level set estimation error do not decrease significantly when $T \geq 5$.




(a)




(b)

Figure 4.2: Performance versus window size $T$

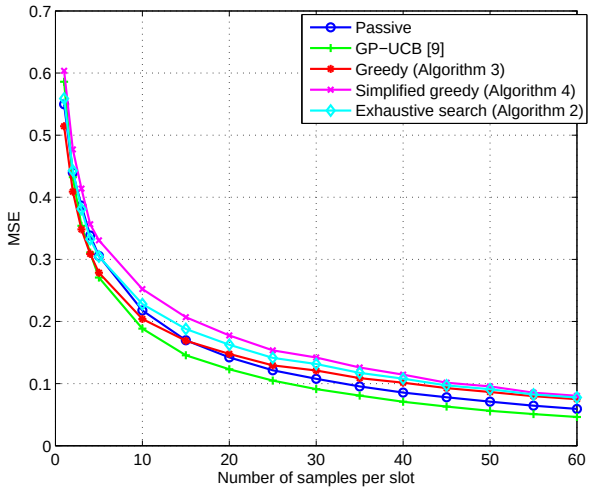
#### 4.7.1.4 Effect of sample size $N$ on the performance

The effects of the number of selected sensing locations on the performance of level set estimation are studied in this example. We set $\rho_t = 0.9$, $\rho_s = 0.9$, $L = 169$ ($d = 13$), and fix the window size $T$ for the sliding window scheme to be 5, i.e., the sensing location selection is

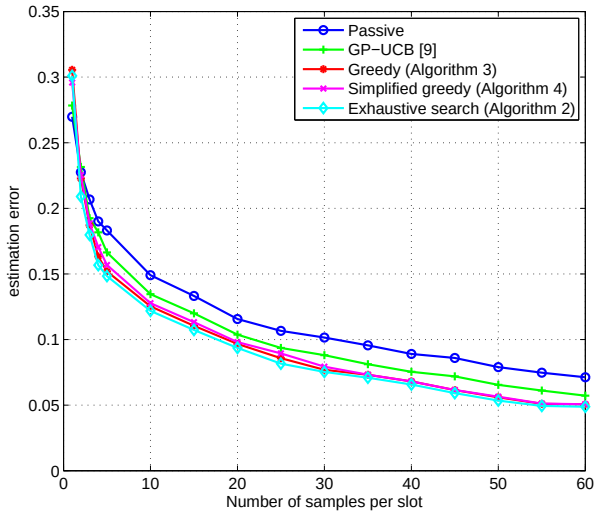
made based on samples collected in the most recent 4 time slots, and the final estimation is based on those plus the new samples collected in the current slot. The sensing location set size $N$ varies from 1 to 60. For each $N$, we randomly generate 1,000 different traces, and the average MSE and level set estimation error are plotted in Figs. 4.3a and 4.3b, respectively.

The MSE and level set error decrease monotonically as $N$ increases for all algorithms. For the level set estimation error, all three proposed algorithms consistently outperform the GP-UCB and the passive algorithm. In addition, the performance of the optimum algorithm with exhaustive search, the greedy algorithm, and the simplified greedy algorithm are almost identical for $N \geq 10$. The performance gap between the our proposed sensing schemes and the passive sensing remains almost a constant (around 0.05) for different values of $N$.

For the MSE, we note that the passive algorithm is slightly worse than the proposed algorithms for small $N$ (less than 10), and it gradually surpasses the greedy algorithms as $N$ increases. The reason that the greedy algorithm has a better MSE performance for small $N$ can be explained by the following fact. The level set estimation error does not only depend on the posterior mean, but also the posterior variance. To minimize the error, it requires small uncertainty in the posterior distribution, which is aligned with minimizing MSE when $N$ is small. As $N$ increases, the posterior mean becomes more important in deciding the level set estimation, as it reflects the locations of the boundary points. Therefore, MSE is no longer the primary goal, and more resources are allocated for searching for the boundaries.




(a)



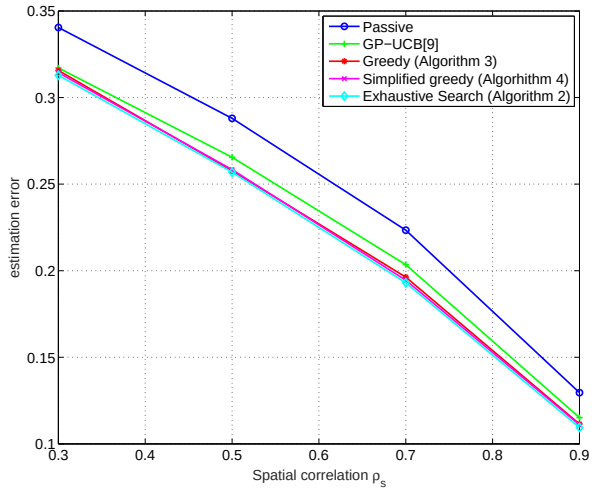

(b)

Figure 4.3: Performance versus sensing set size $N$.

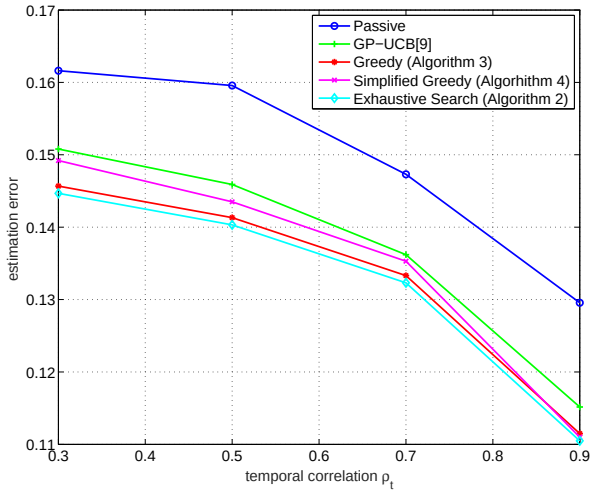
#### 4.7.1.5 Effect of $\rho_s$ and $\rho_t$ on the performance

The effects of spatial correlation coefficient $\rho_s$ and temporal correlation coefficient $\rho_t$ on the performance are studied in this example. We set $L = 169$ ($d = 13$), sample size $N = 15$, and sliding window size $T = 5$. We have $\rho_t = 0.9$ and $\rho_s = 0.9$ in Figs. 4.4a and 4.4b, respectively. The curves are obtained by averaging over 1,000 random trials. As expected, the level set

estimation error is a monotonic decreasing function in both $\rho_s$ or $\rho_t$ for all algorithms. The three proposed algorithms consistently outperform the existing algorithms for all values of $\rho_s$ and $\rho_t$ considered in this example. In Fig. 4.4a, the performance of Algorithms 2, 3, and 4 are almost identical, which again verifies the validity of the approximation in (4.38). Changing the spatial correlation coefficient has a much bigger impact on the estimation performance than changing the temporal correlation coefficient.




(a)




(b)

Figure 4.4: Performance versus spatial correlation $\rho_s$ and temporal correlation $\rho_t$.

#### 4.7.1.6 Computation Complexity

The computation complexities of various algorithms are compared in this example. The complexity is measured by the amount of time required to run one simulation trial. The simulation configurations are the same as those for Fig. 4.2 with $L = 169$ ($d = 13$) and $N = 15$. The simulations are run on a Windows 7 workstation with a 3.10 GHz Intel Core i5-2380P CPU and 16 GB of RAM. The simulation software is Matlab R2011b. The computation time for various algorithms are shown in Fig. 4.5 as a function of the sliding window size $T$. The computation time increases almost linearly with the sliding window size. The passive sensing algorithm has the lowest complexity because it just randomly pick $N$ sensing locations at each time slot, followed by the GP-UCB algorithm and simplified greedy algorithm (Algorithm 4). The greedy algorithm (Algorithm 3) has a slightly higher complexity with additional computation with the off-diagonal elements of $\bar{\mathbf{K}}$. The exhaustive search algorithm (Algorithm 2) has the highest complexity.

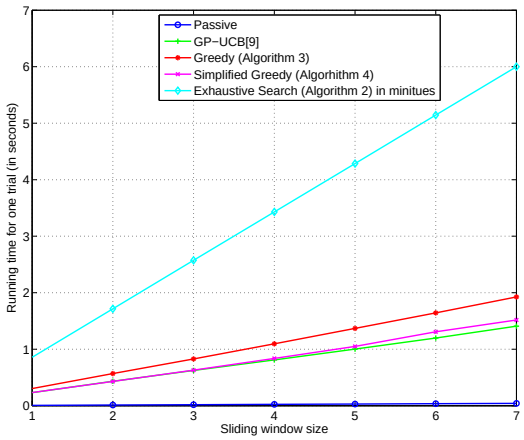



Figure 4.5: Computation time of one trial by various algorithms.

### 4.7.2 Experiment Results

The level set estimations are performed over real world temperature data collected from 200 randomly selected weather stations covering the 48 states of the continental United States. The data are available online at the National Climatic Data Center [20]. We use the daily average temperature data in the month of January from a time span of 14 years (2000 to 2013) as the dataset, thus each location has $T = 31 \times 14 = 434$ time-varying data.

#### 4.7.2.1 Preprossing and Parameter Estimation

Pre-processing is performed over the 434 data samples at each weather station to convert them into a zero-mean random process with unit variance. Denote the raw daily temperature data collected by the $i$-th weather station on the $n$-th day as $x(i, n)$, for $i = 1, \cdots, 200$ and $n = 1, \cdots, 434$. Then the pre-processed data samples are $f(i, n) = \frac{1}{\tilde{\sigma}_i}[x(i, n) - \bar{x}_i]$, where $\bar{x}_i = \frac{1}{T}\sum_{n=1}^{T} x(i, n)$ is the sample mean, and $\tilde{\sigma}_i^2 = \frac{1}{T-1}\sum_{i=1}^{T} |x(i, n) - \bar{x}_i|^2$ is the sample variance of the data collected by the $i$-th weather station.

The spatial and temporal covariance functions of the pre-processed data are modeled with the Matérn covariance function [24],

$$\kappa_{v,l}(r) = \frac{2^{1-v}}{\Gamma(v)} \left( \frac{\sqrt{2v}r}{l} \right)^{v} K_v \left( \frac{\sqrt{2v}r}{l} \right) \tag{4.41}$$

where $v$ and $l$ are the smooth and range parameters. The parameters $(v, l)$ in both the space and time domain, along with the noise variance $\sigma^2$, are estimated jointly through maximum likelihood estimation. For the spatial covariance function, the estimation is performed by using $T = 434$ sets of data in the time domain, and the dimension of each set of data is

$M = 200$. For the temporal covariance function, we are only interested in the temporal covariance within a month of the same year. Therefore, the data in the same month from the same weather station form a 31-dimension data vector, and there are $200 \times 14 = 2800$ sample vectors that are used for the estimation of the temporal covariance function.

We define the hyperparameter vector as $\boldsymbol{\theta} = \{v_s, l_s, v_t, l_t, \sigma^2\}^T$, where $(v_s, l_s)$ are the non-negative smooth and range parameters of the Matérn kernel in the space domain, $(v_t, l_t)$ are the smooth and range parameters in the time domain, and $\sigma^2$ is the variance of the additive noise $\xi$. Define the observation data vector from the $i$-th weather station in one month as $\mathbf{y}_i = [y_i, \cdots, y_{i31}]^T$ with $y_{in}$ being the normalized data sample collected on the $n$-th day of a month by the $i$-th weather station. Stacking the vectors from all 200 stations yields $\mathbf{y} = [\mathbf{y}_1^T, \cdots, \mathbf{y}_{200}^T]^T \in \mathcal{R}^{6200}$. The likelihood function of $\mathbf{y}$ given $\boldsymbol{\theta}$ is

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{(\sqrt{2\pi})^N |\boldsymbol{\Sigma}|} \exp\left\{-\frac{1}{2}\mathbf{y}^T \left(\boldsymbol{\Sigma} + \sigma^2 \mathbf{I}_{6200}\right)^{-1} \mathbf{y}\right\}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix that has a Toeplitz-block-Toeplitz structure, that is, $\boldsymbol{\Sigma}$ is a block Toeplitz matrix, and each sub-matrix is also a Toeplitz matrix. The $(m, n)$-th submatrix of $\boldsymbol{\Sigma}$ is $\kappa_{v_s,l_s}(|m-n|)\mathbf{K}(v_t, l_t) \in \mathcal{R}^{31\times 31}$, for $m, n = 1, \cdots, 200$, where $\mathbf{K}(v_t, l_t)$ is a Toeplitz matrix with the $(i, j)$-th element being $\kappa_{v_t,l_t}(|i-j|)$.

With more observation data in the same month in 14 years, the likelihood function can be written as

$$p(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(14)}|\boldsymbol{\theta}) = \prod_{m=1}^{14} p(\mathbf{y}^{(m)}|\boldsymbol{\theta}),$$

where $\mathbf{y}^{(m)}$ is the 6200-dimension data vector from the $m$-th year. With the maximum likelihood estimation, the estimated parameter vector is $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} p(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(14)} | \boldsymbol{\theta})$.

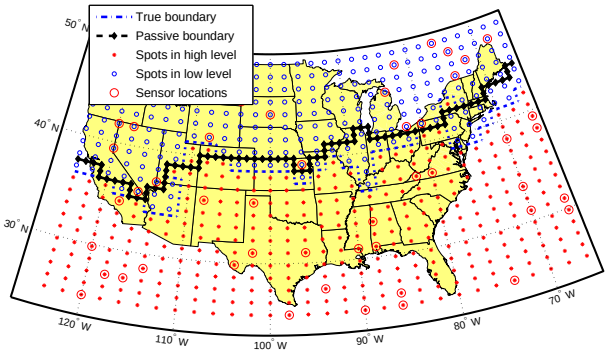
#### 4.7.2.2 Results

The first example is used to illustrate the sensing locations chosen by the proposed algorithms. To visually illustrate the performance of level set estimation, we use a rectangle to cover the US map, and divide the rectangle into segments, with 21 equal-spaced segments in the latitude direction, and 30 equal-spaced segments in the longitude direction as shown in Fig. 4.6a. For a given day, the temperature data of all $L = 21 \times 30 = 630$ segments are obtained by interpolating the 200 weather stations with the Matérn covariance function, and they serve as the ground truth for the experiment. The level set threshold is set to $\gamma = 0.1$.

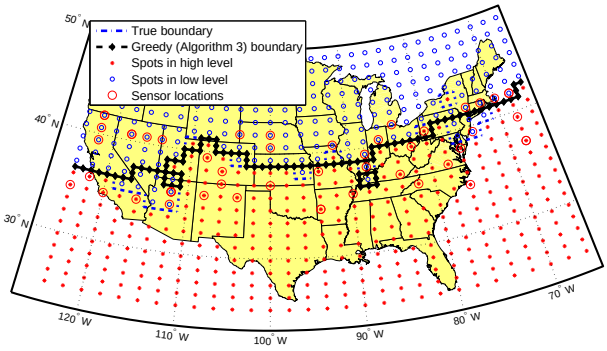
We set $N = 1$ and use the data in one day, and select the first 50 sensing locations with the passive sensing algorithm in Fig. 4.6a and the greedy sensing algorithm (Algorithm 5) in Fig. 4.6b. It can be clearly seen from the two figures that the proposed greedy algorithm selects the sensing locations close to the level set boundary, thus it results in a very accurate estimate of the temperature level set. The level set estimation errors for the passive, GP-UCB, and greedy algorithms are 0.0794, 0.0569, and 0.0429, respectively. On the other hand, the MSE of the passive, GP-UCB, and greedy algorithms are 0.2089, 0.1787, and 0.2213, respectively. Thus the proposed greedy algorithm has the best level set estimation accuracy, even though its MSE performance is the worst.

Then the effect of time window size on the performance is explored in Fig. 4.7. The sensing location set size $N$ is chosen as 30. The time window $T$ varies from 1 to 8. For each $T$, the value is obtained by averaging over all sliding window positions within a month of the

same year, and the results are then averaged again over 14 years. Similar to the simulation results, the level set estimation error of the greedy algorithm and the optimum algorithm with exhaustive search are very close to each other, and they are better than the GP-UCB and the passive algorithms. The level set estimation error of all algorithms do not increase when $T \geq 4$.




(a)




(b)

Figure 4.6: Red circles represent 50 sensing locations selected by the sensing algorithms. The spots in the original truth signal are represented as red solid dots in the high level sets and blue circles in the low level sets.

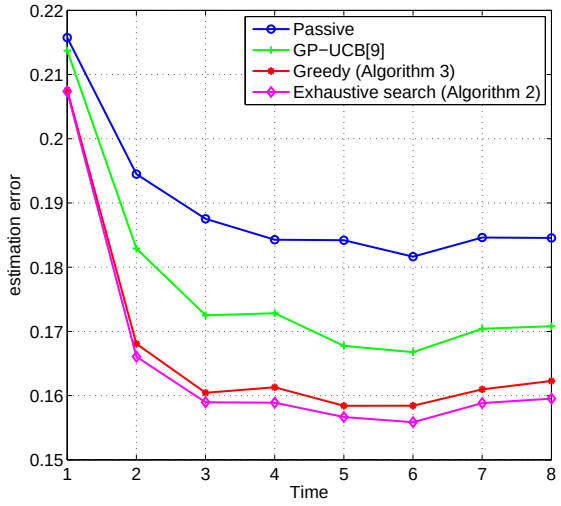



Figure 4.7: Level set error as a function of window size $T$ in the experiment.

Next we study the impacts of sensing set size on system performance in Fig. 4.8. The sliding window size $T$ is chosen as 3. The sensing location set size $N$ varies from 1 to 60. For each $N$, the value is obtained by averaging over all sliding window positions within a month of the same year, and the results are then averaged again over 14 years. Again, the proposed greedy and optimum algorithms consistently outperform the passive and GP-UCB algorithms.

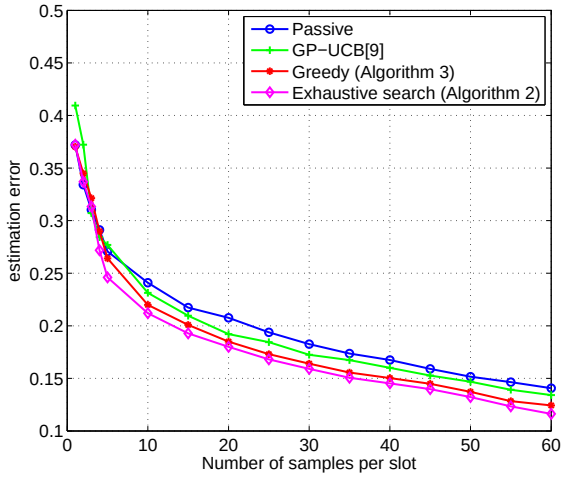



Figure 4.8: Level set error as a function of sensing set size $N$ in the experiment.

## 4.8 Conclusions

We proposed an active sparse sensing scheme for level set estimation in spatial-temporally correlated random field. The sparse sensing scheme can dynamically adjust the selection of sensing locations based on past sensing results, thus achieving the rapid and accurate extraction of level sets in a large random field with a small number of sensing samples. Exact analytical expression of the expected level set estimation error probability were developed by employing an optimum GP regression based level set estimation algorithm. An optimum active sensing algorithm was developed to minimize the level set error probability. Two low complexity greedy algorithms were proposed to minimize an upper bound of the level set error probability. All three algorithms achieved significant performance gains over passive sensing algorithms that do not proactively select the sensing locations, and the GP-UCB algorithm [9] that selects the sensing locations based on a surrogate cost function.

## 4.9 Appendix of the Copyright

### 4.9.1 Copyright Clearance


**Thesis / Dissertation Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

### 4.10 References

[1] Ian F. Akyildiz, Brandon F. Lo, and Ravikumar Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Phys. Commun.*, 4(1):40–62, March 2011.

[2] R. Blasco Serrano, D. Zachariah, D. Sundman, R. Thobaben, and M. Skoglund. A measurement rate-mse tradeoff for compressive sensing through partial support recovery. *IEEE Trans. Sig. Proc.*, 2014.

[3] Jie Chen, Kian Hsiang Low, Colin Keng-Yan Tan, Ali Oran, Patrick Jaillet, John M. Dolan, and Gaurav S. Sukhatme. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. *CoRR*, abs/1206.6230, 2012.

[4] Edward Y Chow and Alan S Willsky. Bayesian design of decision rules for failure detection. *IEEE Trans. Aerospace and Electronic Systems*, AES-20(6):761–774, 1984.

[5] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[6] John W Craig. A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations. In *Military Communications Conference, 1991. MILCOM'91, Conference Record, Military Communications in a Changing World., IEEE*, pages 571–575. IEEE, 1991.

[7] Benjamin Friedlander and B. Porat. The exact cramer-rao bound for gaussian autoregressive processes. *IEEE Trans. Aerospace and Electronic Systems*, 25(1):3–7, Jan 1989.

[8] Google Spectrum Database. available online at http://www.google.org/spectrum/whitespace/channel/.

[9] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, IJCAI'13, pages 1344–1350, 2013.

[10] Dongbing Gu and Huosheng Hu. Spatial gaussian process regression with mobile sensor networks. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(8):1279–1290, 2012.

[11] Xiao Han, Chenyang Xu, and Jerry L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.

[12] M. Hurtado, C.H. Muravchik, and A. Nehorai. Enhanced sparse bayesian learning via statistical thresholding for signals in structured noise. *IEEE Trans. Sig. Proc.*, 61(21):5430–5443, Nov 2013.

[13] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[14] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, AAAI'07, pages 1650–1654. AAAI Press, 2007.

[15] Andreas Krause and Carlos Guestrin. Nonmyopic active learning of gaussian processes: An exploration-exploitation approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 449–456, New York, NY, USA, 2007. ACM.

[16] David J. C. MacKay. Information-based objective functions for active data selection. *Neural Comput.*, 4(4):590–604, July 1992.

[17] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[18] Nicolas Maisonneuve, Matthias Stevens, and Bartek Ochab. Participatory noise pollution monitoring using mobile phones. *Info. Pol.*, 15(1,2):51–71, April 2010.

[19] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 323–336, New York, NY, USA, 2008. ACM.

[20] National Climatic Data Center. Quality Controlled Local Climatological Data (QCLCD),available online at http://www.ncdc.noaa.gov/.

[21] J. Paisley, Xuejun Liao, and L. Carin. Active learning and basis selection for kernel-based linear models: A bayesian perspective. *IEEE Trans. Sig. Proc.*, 58(5):2686–2700, May 2010.

[22] Eulogio Pardo-Igúzquiza. Maximum likelihood estimation of spatial covariance parameters. *Mathematical Geology*, 30(1):95–108, 1998.

[23] H Van Dyke Parunak, Sven Brueckner, and James J Odell. Swarming coordination of multiple uav's for collaborative sensing. In *Proceedings of Second AIAA" Unmanned Unlimited" Systems, Technologies, and Operations Conference*, 2003.

[24] Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[25] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[26] L.L. Scharf and D. Lytle. Stability of parameter estimates for a gaussian process. *IEEE Trans. Aerospace and Electronic Systems*, AES-9(6):847–851, Nov 1973.

[27] C. Scott and M. Davenport. Regression level set estimation via cost-sensitive classification. *IEEE Trans. Sig. Proc.*, 55(6):2752–2757, June 2007.

[28] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[29] Aarti Singh, Robert Nowak, and Parmesh Ramanathan. Active learning for adaptive mobile sensing networks. In *Proceedings of the 5th international conference on Information processing in sensor networks*, IPSN '06, pages 60–68, New York, NY, USA, 2006. ACM.

[30] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[31] Zuoen Wang, Jingxian Wu, Jing Yang, and Hai Lin. Energy-efficient wireless sensing for level set estimations. *Access, IEEE*, 3:1480–1490, 2015.

[32] Zuoen Wang, Jingxian Wu, Jing Yang, and Hai Lin. Optimum level set estimation of a time-varying random field under a power constraint. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[33] Zuoen Wang, Jingxian Wu, Jing Yang, and Hai Lin. Optimal energy efficient level set estimation of spatially-temporally correlated random fields. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[34] Richard Webster and Oliver A. Margaret. *Statistical methods in soil and land resource survey*. Oxford University Press, 1990.

[35] R. M. Willett and R. D. Nowak. Minimax optimal level-set estimation. *IEEE Trans. Img. Proc.*, 16(12):2965–2979, December 2007.

[36] Jing Yang, Zuoen Wang, and Jingxian Wu. Level set estimation with dynamic sparse sensing. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 487–491. IEEE, 2014.

# Chapter 5

# Optimum Distributed Estimation of a Spatially Correlated Random Field

## 5.1 Abstract

We study the distributed estimation of a spatially correlated random field with decentralized sensor networks. Nodes in the network take spatial samples of the random field, then each node estimates the values of arbitrary points on the random field by iteratively exchanging information with each other. The objective is to minimize the estimation mean squared error (MSE) while ensuring all nodes reach a distributed consensus on the estimation results. We propose a distributed iterative linear minimum mean squared error (LMMSE) algorithm that defines an information propagation stage and a local estimation stage in each iteration. The key parameters of the algorithm, including an edge weight matrix and a sample weight matrix, are designed to minimize an MSE upper bound at all nodes when the number of iterations is large. It is shown that the optimum performance can be achieved by distributively projecting the high dimension measurement samples from all nodes into a low dimension subspace related to the covariance matrices of data and noise samples, and this projection is achieved in a distributed manner through iterative information propagation. The low dimension projection can significantly reduce the amount of data exchanged in the network, thus improve the convergence speed of the iterative algorithm. Simulation and experimental results show that all nodes in a large network can obtain accurate estimation results with less iterations and lower complexities than existing algorithms.

## 5.2 Introduction

Wireless sensor networks (WSNs) have a wide range of applications, such as precision agriculture, environment monitoring, surveillance, and disaster relief, etc. In many of the WSN applications, the sensors are used to monitor a physical quantity that can be modeled as a spatially correlated random field, such as temperature, vibration, pressure, and pollutant concentration, etc [23]. Sensor nodes take samples of the random field at their respective locations, and the results are then used to estimate the values of arbitrary points on the random field.

The WSN can be classified into two categories, centralized and decentralized WSNs. In a centralized WSN, the measurement results from all nodes are congregated at a fusion center (FC), either through direct transmission or by using other nodes as relays. Information processing is performed centrally at the FC. Centralized WSNs are easy to design, but they also suffer from a lot of limitations such as high cost of the FC, communication bottlenecks at areas close to the FC, and susceptible to node failures, etc. These problems can be easily addressed by a decentralized WSN, where the information processing is performed at each sensor node in a distributed manner without the need of a central controller. The information processing is performed collaboratively among the nodes through iterative information exchange among neighboring nodes.

Distributed estimation is one of the most fundamental collaborative information processing problems in distributed WSNs, where the nodes distributively perform estimation of certain physical quantities through information exchange [1–5, 7, 8, 11–13, 15, 18, 20–22]. Most distributed estimation algorithms involve two components: a local estimator and a

distributed consensus algorithm that can be used to improve the estimation performance. Algorithms based on distributed Kalman filters are presented in [12, 13] by using knowledge of the states and observation models of the underlying data. When the measurements are spatially correlated, a distributed Kriged Kalman filter is proposed to obtain weighted least squares estimates at the nodes [4]. A distributed maximum a posteriori (MAP) estimator is presented in [11] for multi-robot cooperative localization, where a distributed conjugate gradient algorithm is employed to reduce the computation complexity. In [21], a distributed Bayesian estimation algorithm is proposed to estimate unknown parameters of a linear model with unknown observation covariances.

In addition to estimation algorithms, distributed consensus algorithms [1, 3, 6, 14, 19, 25] are used to enforce agreement among cooperating nodes. In a consensus procedure, each node maintains a state, shares its state with its neighboring nodes, and updates the state by using information from its neighbors. An agreement about the state can be reached by all nodes in the network through iterative information exchange. In [25], a simple scheme of distributed average consensus was proposed to compute the maximum-likelihood estimate of system parameters based on noisy linear measurements of the parameters. A distributed MAP and a distributed linear minimum mean squared error (D-LMMSE) algorithm are proposed in [1], where a set of bridge nodes is introduced to enable information consensus. In [3], the distributed estimation and consensus is achieved by using a diffusion recursive least squares (D-RLS) algorithm, which exploits more network connectivity and achieves faster network convergence compared to the incremental RLS proposed in [17].

This paper proposes a new distributed iterative LMMSE algorithm for the estimation of a spatially correlated random field. A group of distributed sensor nodes take spatial

samples of the random field, and the nodes need to estimate the values of some arbitrary points on the random field in a distributed manner through iterative information exchange. The objective is to minimize worst case mean squared error (MSE) at all nodes, while ensuring all nodes reach a distributed consensus of the estimation results. The algorithm performs two actions in each iteration: information propagation and local estimation. In the information propagation stage, a node updates a locally maintained state vector by using a linear combination of its own information and state vectors sent out by its neighbors in the previous iteration. The updated state vector is then broadcast to all its neighbors. In the local estimation stage, each sensor performs iterative LMMSE by using its updated state vector and its own estimates from the previous iteration. The initial state vector is obtained by multiplying the measurement sample with a coefficient vector.

The proposed algorithm requires two key parameters, an edge weight matrix for state vector updates and exchanges, and a sample weight matrix for constructing the initial state vector. The two matrices are designed to minimize an MSE upper bound while ensuring distributed consensus when the number of iterations is large. It is found that the optimum design is equivalent to projecting the high dimension samples from all nodes into a low dimension subspace that is related to the covariance matrices of data and noise. The projection is performed in a distributed manner through iterative information propagation, with the help of the edge weight and sample weight matrices. The low dimension projection can significantly reduce the among of information exchanged in the network, and achieve faster convergence compared to existing algorithms such as D-LMMSE [1] and D-RLS [3]. A distributed learning algorithm is also presented for the distributed estimation of the spatial covariance that is necessary for the implementation of the proposed algorithm.

The remainder of this paper is organized as follows. The problem is formulated in Section II. The distributed iterative LMMSE algorithm is presented in Section III. Section IV discusses the optimum design of the edge weight matrix and sample weight matrix. Section V proposes a distributed learning algorithm for spatial covariance estimation. Simulation results are given in Section VI and Section VII concludes the paper.

### 5.3 Problem formulation

Consider a sensor network with $n$ sensor nodes, which are used to monitor a spatially correlated random field $\mathcal{F}$. Denote the spatial coordinate of the $i$-th sensor node as $\mathbf{c}_i \in \mathcal{F}$. The sensor network can be represented as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, n\}$ is the set of nodes and $\mathcal{E}$ is the set of edges, with $\{i, j\} \in \mathcal{E}$ connecting nodes $i$ and $j$. The set of neighbors of node $i$ is denoted $\mathcal{N}_i = \{j : \{i, j\} \in \mathcal{E}\}$. Two nodes can directly exchange information with each other if they are neighbors. A graph is called strongly connected (SC) if there is a path connecting any two nodes in the graph. The sensor network can be modeled as an unweighted SC graph.

The sensor network is used to monitor a physical quantity $x(\mathbf{c}) \in \mathcal{R}$ in the random field, such as the temperature, humidity, or the normalized difference vegetation index (NDVI) in an agricultural field. The quantity $x(\mathbf{c})$ is assumed to be of zero mean, which can be obtained by subtracting the mean of a non-zero mean process. The physical quantity $x(\mathbf{c})$ has a spatial correlation $r_{xx'} = \mathbb{E}[x(\mathbf{c})x(\mathbf{c}')]$. It is assumed that the random filed is spatially wide-sense stationary with $r_{xx'} = k_s(\|\mathbf{c} - \mathbf{c}'\|)$, where $k_s(\cdot)$ is a covariance kernel function, and $\|\mathbf{c} - \mathbf{c}'\|$ is the Euclidean distance between the coordinates $\mathbf{c}$ and $\mathbf{c}'$.

Each sensor node can obtain a measurement of the physical quantity at its own location.

The measurement taken by the $i$-th sensor is

$$z(\mathbf{c}_i) = x(\mathbf{c}_i) + \epsilon(\mathbf{c}_i). \tag{5.1}$$

where $\epsilon(\mathbf{c}_i)$ is the sensing noise with variance $\sigma^2$. Denote the measurement vector and the corresponding data vector as $\mathbf{z} := [z(\mathbf{c}_1), \cdots, z(\mathbf{c}_n)]^T \in \mathcal{R}^n$ and $\mathbf{x} := [x(\mathbf{c}_1), \cdots, x(\mathbf{c}_n)]^T \in \mathcal{R}^n$, respectively, where $\mathbf{a}^T$ represents matrix transpose.

In the distributed WSN, each node will estimate the values of $k$ locations in the random field, and denote them as $\mathbf{s} := [x(\mathbf{c}'_1), x(\mathbf{c}'_2), \ldots, x(\mathbf{c}'_k)]^T$. The estimation will be performed by each node separately in a distributed manner by using its own measurement and information received from its neighboring nodes. It should be noted that $\mathbf{s}$ could be the same or different from the measurement vector $\mathbf{x}$.

In a decentralized network, the nodes need to exchange information with each other in order to obtain an estimate of $\mathbf{s}$. Without loss of generality, the time is divided into slots. In each slot, a node can update its estimate of $\mathbf{s}$ by using information received from all of its neighbors. Denote the estimate from node $i$ at slot $t$ as $\hat{\mathbf{s}}(i,t) \in \mathcal{R}^n$, for $i = 1, \cdots, n$ and $t = 0, 1, 2, \cdots$.

The design objective is to find the information exchange scheme such that all nodes in the network can reach a consensus and obtain an accurate estimate of $\mathbf{s}$. Denote the estimation mean squared error (MSE) at the $i$-th node and time slot $t$ as $\sigma_e^2(i,t) = \mathbb{E}\left[\|\hat{\mathbf{s}}(i,t) - \mathbf{s}\|^2\right]$,

where $\|\mathbf{a}\| = \sqrt{\mathbf{a}^T\mathbf{a}}$ is the $\ell_2$-norm of the vector $\mathbf{a}$. Then the problem can be formulated as

$$\begin{aligned} \text{(P1)} \quad & \text{minimize} \quad \lim_{t\to\infty} \max_i \sigma^2_{\mathbf{e}}(i,t) \\ & \text{subject to} \quad \lim_{t\to\infty} |\hat{\mathbf{s}}(i,t) - \hat{\mathbf{s}}(j,t)| = 0, \forall i \neq j \end{aligned} \tag{5.2}$$

The constraint in (P1) is used to ensure that all nodes in the network reach a consensus on the estimation results.

The optimization problem can be easily solved in a centralized system, where the information collected by all the nodes are congregated at a FC. In this case, the FC has full knowledge of the measurement vector $\mathbf{z}$ and it can obtain an optimum estimate of $\mathbf{s}$ by using the LMMSE estimator as

$$\hat{\mathbf{s}}_{\text{FC}} = \mathbf{R}_{\mathbf{sx}}(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)^{-1}\mathbf{z} \tag{5.3}$$

where $\mathbf{R}_{\mathbf{ab}} = \mathbb{E}[\mathbf{a}\mathbf{b}^T]$ is the cross-correlation matrix between vectors $\mathbf{a}$ and $\mathbf{b}$, and $\mathbf{I}_n$ is a size-$n$ identity matrix. The covariance matrices, $\mathbf{R}_{\mathbf{sx}}$ and $\mathbf{R}_{\mathbf{xx}}$, are generally unknown to the nodes in the network, but they can be acquired either during a training phase or from the underlying physics of the quantity of interest. A distributed covariance learning algorithm is presented in Section 5.6, where the nodes can learn the covariance in a distributed manner by exchanging data samples with their immediate neighbors. Since the covariance changes much slower compared to the actual values $\mathbf{x}$ and $\mathbf{s}$, the training only needs to be performed once at the beginning.

In a decentralized network, the information of $\mathbf{z}$ is distributed at all the nodes and no

node has full knowledge of $\mathbf{z}$. Each node performs the estimation individually by using its own measurement and information exchanged from its neighbors. This necessitates the design of distributed estimation algorithms.

## 5.4 Distributed iterative LMMSE estimation

In this section we propose a distributed iterative LMMSE estimation algorithm for the WSN. The algorithm is iterative in the sense that each node will exchange information with its neighbors in an iterative manner until convergence.

In the iterative algorithm, the time is divided into slots, and each time slot corresponds to one iteration of the algorithm. Each node maintains a size-$m$ state vector $\mathbf{y}(i,t) \in \mathcal{R}^m$ with $m \leq n$. The state vector is updated in each iteration and then shared with all of its neighboring nodes.

In the $t$-th iteration, each node will perform two actions:

1. Information Propagation: node $i$ will update its state vector $\mathbf{y}(i,t)$, and broadcast it to all its neighboring nodes with indices $j \in \mathcal{N}_i$.

2. Local Estimation: node $i$ will obtain an estimate of $\mathbf{s}$ by using $\mathbf{y}(i,t)$, and denote the estimation result as $\hat{\mathbf{s}}(i,t)$.

We will discuss the details of the two stages in the next two subsections, respectively.

### 5.4.1 Iterative Information Propagation

In the initial iteration with $t = 0$, the vector broadcast by the $i$-th node is $\mathbf{y}(i,0) = \tilde{\mathbf{v}}_i z(\mathbf{c}_i)$, where $z(\mathbf{c}_i)$ is the scalar measurement at the $i$-th sensor, and the weight vector $\tilde{\mathbf{v}}_i \in \mathcal{R}^m$ is

designed to minimize the estimation MSE required for distributed consensus. The optimum design of $\tilde{\mathbf{v}}_i$ will be discussed in the next section. The initial state vector $\mathbf{y}(i,0)$ contains $m$ weighted copies of the $i$-th node's measurement $z(\mathbf{c}_i)$.

In the $(t+1)$-th iteration, node $i$ will first update its state vector by using its own state vector and state vectors received from all its neighbors in the previous iteration, as

$$\mathbf{y}(i,t+1) = w_{ii}\mathbf{y}(i,t) + \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{y}(j,t), i = 1,2,\ldots,n, \tag{5.4}$$

where $t = 0, 1, \cdots$, and $w_{ij}$ is a weight coefficient applied by node $i$ to the vector received from node $j \in \mathcal{N}_i$. We have $w_{ij} = 0$ if $j \notin \mathcal{N}_i$. A weight matrix $\mathbf{W} \in \mathcal{R}^{n \times n}$ is formed by setting its element on the $i$-th row and $j$-th column as $(\mathbf{W})_{ij} = w_{ij}$, if $j \in \mathcal{N}_i$ and $(\mathbf{W})_{ij} = 0$ otherwise. The matrix $\mathbf{W}$ is denoted as the edge weight matrix. The edge weight matrix $\mathbf{W}$ is designed to ensure the convergence of the iterative process, and the optimum design of $\mathbf{W}$ will be discussed in the next section.

The updated state vector $\mathbf{y}(i,t+1)$ will then be broadcast to all its neighbors with indices $j \in \mathcal{N}_i$.

We can form a matrix by using the vectors from all nodes in the networks as $\mathbf{Y}(t) = [\mathbf{y}(1,t), \mathbf{y}(2,t), \ldots, \mathbf{y}(n,t)]^T \in \mathcal{R}^{n \times m}$, then we can rewrite (5.4) in a compact form as

$$\mathbf{Y}(t+1) = \mathbf{W}\mathbf{Y}(t), \quad \text{for } t = 0, 1, \cdots \tag{5.5}$$

The initial matrix $\mathbf{Y}(0)$ can be alternatively represented as

$$\mathbf{Y}(0) = \mathbf{Z}_d\mathbf{V} \tag{5.6}$$

where $\mathbf{Z}_d = \text{diag}\{\mathbf{z}\} \in \mathcal{R}^{n\times n}$ is a diagonal matrix with $\mathbf{z}$ on its main diagonal, and $\mathbf{V} = [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_n]^T = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m] \in \mathcal{R}^{n\times m}$ with $\mathbf{v}_k \in \mathcal{R}^n$, for $k = 1, \cdots, m$. We denote $\mathbf{V}$ as the sample weight matrix because its elements are directly applied to the measurement samples. Combining (5.5) and (5.6) yields

$$\mathbf{Y}(t) = \mathbf{W}^t\mathbf{Y}(0) = \mathbf{W}^t\mathbf{Z}_d\mathbf{V}, \tag{5.7}$$

where $\mathbf{W}^t$ is the $t$-th power of the matrix $\mathbf{W}$.

We can obtain the state vector at the $i$-th node at time slot $t$ by extracting the $i$-th row of $\mathbf{Y}(t)$ as

$$\mathbf{y}^T(i,t) = \mathbf{w}_i(t)\mathbf{Z}_d\mathbf{V} \tag{5.8}$$

where $\mathbf{w}_i(t)$ is the $i$-th row of the matrix $\mathbf{W}^t$. Define a diagonal matrix $\mathbf{W}_i(t) = \text{diag}\{\mathbf{w}_i^T(t)\} \in \mathcal{R}^{n\times n}$, then (5.8) can be alternatively expressed as

$$\mathbf{y}(i,t) = \mathbf{A}(i,t)\mathbf{z} \in \mathcal{R}^m \tag{5.9}$$

where $\mathbf{A}(i,t) = \mathbf{V}^T\mathbf{W}_i(t) \in \mathcal{R}^{m\times n}$.

### 5.4.2 Iterative LMMSE Estimation

At the $t$-th time slot, node $i$ will obtain an estimate of $\mathbf{s}$ by using the aggregated state vectors $\{\mathbf{y}(i,t')\}_{t'=0}^{t}$, and the estimated vector is denoted as $\hat{\mathbf{s}}(i,t)$. Without loss of generality, we will focus on the operation at node $i$. To simplify notation, we will skip the index $i$ in subsequent discussions as

$$\mathbf{y}_t := \mathbf{y}(i,t),\ \mathbf{s}_t := \mathbf{s}(i,t),\ \mathbf{A}_t := \mathbf{A}(i,t) \tag{5.10}$$

Thus (5.9) can be alternatively written as

$$\mathbf{y}_t = \mathbf{A}_t\mathbf{z} \tag{5.11}$$

The aggregated state vector $\mathbf{y}_{1:t} := [\mathbf{y}_1^T, \ldots, \mathbf{y}_t^T]^T$ can be represented as

$$\mathbf{y}_{1:t} = \mathbf{A}_{1:t}\mathbf{z} = \mathbf{A}_{1:t}(\mathbf{x} + \boldsymbol{\epsilon}). \tag{5.12}$$

where $\boldsymbol{\epsilon} := [\epsilon(\mathbf{c}_1), \cdots, \epsilon(\mathbf{c}_n)]^T$ and $\mathbf{A}_{1:t} = [\mathbf{A}_1^T, \cdots, \mathbf{A}_t^T]^T \in \mathcal{R}^{mt\times n}$.

We have the following results regarding the LMMSE estimator.

*Lemma 1*: Given $\mathbf{y}_{1:t}$, the LMMSE estimate of $\mathbf{s}$ at node $i$ and time slot $t$ is

$$\hat{\mathbf{s}}(t) = \mathbf{R}_{\mathbf{sx}}\mathbf{A}_{1:t}^T\left[\mathbf{A}_{1:t}(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)\mathbf{A}_{1:t}^T\right]^{-}\mathbf{y}_{1:t} \tag{5.13}$$

where $\mathbf{A}^{-}$ is the general inverse of the matrix $\mathbf{A}$.

**Proof:** The proof is in Appendix 5.9.1. ■

The memory requirement and computation complexity of (5.13) increases with time $t$, and they become prohibitively high when $t$ and/or $n$ is large. Given the iterative nature of the distributed consensus algorithm, we propose to calculate $\hat{\mathbf{s}}(t)$ iteratively. The iterative LMMSE estimation algorithm is presented in Theorem 5.1 below.

To facilitate analysis and presentation, we first define the following notations that will be used in Theorem 5.1. For any arbitrary random vector $\mathbf{p}$, denote the LMMSE estimator of $\mathbf{p}$ given samples history $\mathbf{y}_{1:t}$ at time slot $t$ as

$$\hat{\mathbf{p}}(t) = \mathbb{E}[\mathbf{p}|\mathbf{y}_{1:t}] = \mathbf{R}_{\mathbf{p}\mathbf{y}_{1:t}}\mathbf{R}^{-}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}}\mathbf{y}_{1:t} \tag{5.14}$$

where $\mathbf{p}$ could be $\mathbf{z}$, $\mathbf{s}$, or $\mathbf{y}_{t+1}$

Define the cross-covariance matrix between two LMMSE estimates $\hat{\mathbf{p}}(t)$ and $\hat{\mathbf{q}}(t)$ as

$$\mathbf{C}_{\mathbf{pq}}(t) = \mathbb{E}[(\hat{\mathbf{p}}(t) - \mathbf{p})(\hat{\mathbf{q}}(t) - \mathbf{q})^T] \tag{5.15}$$

where $\mathbf{p}$ and $\mathbf{q}$ are the true values of $\hat{\mathbf{p}}(t)$ and $\hat{\mathbf{q}}(t)$, respectively.

With the above notations, we have the following theorem that shows how to iteratively update $\hat{\mathbf{s}}(t)$ by using $\hat{\mathbf{s}}(t-1)$ and $\mathbf{y}_t$.

*Theorem 5.1*: (Iterative LMMSE) The LMMSE estimation $\hat{\mathbf{s}}(t)$ given sample history $\mathbf{y}_{1:t}$ at node $i$ and time slot $t$ (c.f. Lemma 1) can be iteratively calculated as

$$\hat{\mathbf{s}}(t) = \hat{\mathbf{s}}(t-1) + \mathbf{C}_{\mathbf{sz}}(t-1)\mathbf{A}_t^T\mathbf{B}_t^{-}\big(\mathbf{y}_t - \mathbf{A}_t\hat{\mathbf{z}}(t-1)\big) \tag{5.16}$$

where $\hat{\mathbf{z}}(t) \in \mathcal{R}^n$ can be iteratively calculated as

$$\hat{\mathbf{z}}(t) = \hat{\mathbf{z}}(t-1) + \mathbf{C}_{\mathbf{zz}}(t-1)\mathbf{A}_t^T \mathbf{B}_t^- \left(\mathbf{y}_t - \mathbf{A}_t \hat{\mathbf{z}}(t-1)\right) \tag{5.17}$$

and $\mathbf{B}_t = \mathbf{A}_t \mathbf{C}_{\mathbf{zz}}(t-1)\mathbf{A}_t^T \in \mathcal{R}^{m \times m}$.

The auto- and cross-covariance matrices can be iteratively updated as

$$\mathbf{C}_{\mathbf{zz}}(t) = \mathbf{C}_{\mathbf{zz}}(t-1) - \mathbf{C}_{\mathbf{zz}}(t-1)\mathbf{A}_t^T \mathbf{B}_t^- \mathbf{A}_t \mathbf{C}_{\mathbf{zz}}(t-1) \tag{5.18}$$

$$\mathbf{C}_{\mathbf{sz}}(t) = \mathbf{C}_{\mathbf{sz}}(t-1) - \mathbf{C}_{\mathbf{sz}}(t-1)\mathbf{A}_t^T \mathbf{B}_t^- \mathbf{A}_t \mathbf{C}_{\mathbf{zz}}(t-1) \tag{5.19}$$

$$\mathbf{C}_{\mathbf{ss}}(t) = \mathbf{C}_{\mathbf{ss}}(t-1) - \mathbf{C}_{\mathbf{sz}}(t-1)\mathbf{A}_t^T \mathbf{B}_t^- \mathbf{A}_t \mathbf{C}_{\mathbf{zs}}(t-1) \tag{5.20}$$

with the initial states $\mathbf{C}_{\mathbf{zz}}(0) = \mathbf{R}_{\mathbf{zz}} = \mathbf{R}_{\mathbf{xx}} + \sigma^2 \mathbf{I}_n \in \mathcal{R}^{n \times n}$, $\mathbf{C}_{\mathbf{sz}}(0) = \mathbf{R}_{\mathbf{sx}} \in \mathcal{R}^{k \times n}$ and $\mathbf{C}_{\mathbf{ss}}(0) = \mathbf{R}_{\mathbf{ss}} \in \mathcal{R}^{k \times k}$.

**Proof:** The proof is in Appendix 5.9.2. ■

Theorem 5.1 provides an iterative implementation of the LMMSE in Lemma 1. The iterative procedures described in (5.16)-(5.20) in Theorem 5.1 can obtain the same estimation results as in Lemma 1, but with a fixed memory requirement and much less complexity.

**Comment 1:** The memory requirement of the iterative LMMSE algorithm in Theorem 5.1 does not change with time $t$. At time $t$, the iterative LMMSE needs to store the vectors $\hat{\mathbf{s}}(t) \in \mathcal{R}^k$, $\hat{\mathbf{z}}(t) \in \mathcal{R}^n$, and the matrices $\mathbf{A}_t \in \mathcal{R}^{m \times n}$, $\mathbf{C}_{\mathbf{zz}}(t) \in \mathcal{R}^{n \times n}$, $\mathbf{C}_{\mathbf{sz}}(t) \in \mathcal{R}^{k \times n}$, and $\mathbf{C}_{\mathbf{ss}}(t) \in \mathcal{R}^{k \times k}$. The total memory required by the iterative LMMSE is thus on the order of $k + n + mn + kn + n^2 + k^2$, which is independent of $t$.

**Comment 2:** The computation complexity of the iterative LMMSE algorithm in Theorem 5.1 does not change with time $t$. Since all vectors and matrices in (5.16)-(5.20) are of fixed sizes, the complexities of the iterative calculations described in (5.16)-(5.20) are also fixed at each iteration, and it does not change with respect to $t$.

**Comment 3:** In the distributed iterative LMMSE algorithm, the nodes only exchange information about state vectors $\mathbf{y}(t)$, which are weighted linear combinations of the measurement samples at different nodes. The estimated vectors $\hat{\mathbf{s}}(t)$ are not exchanged among the nodes. This is different from most existing distributed estimation algorithm, such as the D-RLS [3] and incremental RLS [17], which exchange both measurement samples and estimation results. Thus the proposed algorithm requires less information exchange than the incremental RLS or diffusion RLS algorithm. We will show later through simulations that the proposed algorithm also converges faster than those RLS-based algorithms.

The results in Theorem 5.1 indicate that the estimation of $\mathbf{s}$ at each node can be iteratively updated as new information is received from its neighbors. The expected MSE of the LMMSE estimator $\hat{\mathbf{s}}(t)$ at node $i$ and time slot $t$ is

$$\sigma_{\mathbf{e}}^2(i,t) = \frac{1}{k}\text{trace}(\mathbf{C}_{\mathbf{ss}}(t)) \tag{5.21}$$

For the special case when $n = m$, we have the following result regarding the LMMSE estimation when the number of iterations is larger than the diameter of the network $d$, which is defined as the shortest distance between the two most distant nodes in the network.

*Corollary 5.1*: Consider the special case that $m = n$. Assume the sample weight matrix $\mathbf{V}$ is of full column rank, and this can be easily achieved by design. When the number of

iterations $t$ is no less than the diameter of the network $d$, then $\hat{\mathbf{s}}(t) = \hat{\mathbf{s}}(d)$, $\forall t \geq d$, and all nodes have the same estimate $\hat{\mathbf{s}}(t)$.

**Proof:** The proof is in Appendix 5.9.3. ■

Corollary 5.1 means that the distributed iterative LMMSE algorithm converges in at most $d$ iterations if $m = n$ with a full rank sample weight matrix $\mathbf{V}$, and all nodes reach distributed consensus in at most $d$ iterations. We will perform optimum designs of both $\mathbf{W}$ and $\mathbf{V}$ when $m < n$ in the next section.

### 5.4.3 Performance Upper Bound

The MSE result in (5.21) is obtained by performing LMMSE by using the aggregated state vector $\mathbf{y}_{1:t}$. It is expressed as a function of the edge weight matrix $\mathbf{W}$ and the sample weight matrix $\mathbf{V}$ through the matrix $\mathbf{A}_t = \mathbf{A}(i,t) = \mathbf{V}^T\mathbf{W}_i(t)$, where $\mathbf{W}_i(t)$ is a diagonal matrix with its diagonal being the $i$-th row of $\mathbf{W}^t$. The MSE can be calculated numerically through iterations as shown in Theorem 5.1. However, the iterative calculation does not provide a closed-form expression of the MSE in the form of $\mathbf{W}$ and $\mathbf{V}$, which need to be designed to solve the optimization problem. To address this issue, we propose to design the matrices $\mathbf{W}$ and $\mathbf{V}$ by using an upper bound of $\sigma^2_{\mathbf{e}}(i,t)$, which is obtained by performing LMMSE estimation with only the current state vector $\mathbf{y}_t$.

*Corollary 5.2*: Given $\mathbf{y}_t$, the LMMSE estimate of $\mathbf{s}$ at node $i$ and time slot $t$ is

$$\tilde{\mathbf{s}}(i,t) = \mathbf{R}_{\mathbf{sx}}\mathbf{A}_t^T \left[\mathbf{A}_t(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)\mathbf{A}_t^T\right]^- \mathbf{y}_t. \tag{5.22}$$

The corresponding error covariance matrix $\tilde{\mathbf{C}}_{\mathbf{ss}}(t) = \mathbb{E}[(\tilde{\mathbf{s}}(i,t) - \mathbf{s}_t)(\tilde{\mathbf{s}}(i,t) - \mathbf{s}_t)^T]$ is

$$\tilde{\mathbf{C}}_{\mathbf{ss}}(t) = \mathbf{R}_{\mathbf{ss}} - \mathbf{R}_{\mathbf{sx}}\mathbf{A}_t^T \left[\mathbf{A}_t(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)\mathbf{A}_t^T\right]^- \mathbf{A}_t\mathbf{R}_{\mathbf{sx}}^T \tag{5.23}$$

The average MSE at node $i$ and time slot $t$ is thus

$$\tilde{\sigma}_{\mathbf{e}}^2(t,i) := \frac{1}{k}\text{trace}(\tilde{\mathbf{C}}_{\mathbf{ss}}(t)) \tag{5.24}$$

which is an upper bound of $\sigma_{\mathbf{e}}^2(i,t)$ in (5.21).

**Proof:** The proof is in Appendix 5.9.4. ■

## 5.5 Asymptotically optimum Design of the distributed algorithm

In this section, we perform the asymptotically optimum design of the distributed estimation algorithm when $t$ is large. The distributed estimation algorithm presented in the previous section depends on two matrices, the edge weight matrix $\mathbf{W}$, and the sample weight matrix $\mathbf{V}$. We will study the design of $\mathbf{W}$ and $\mathbf{V}$ that can minimize the MSE upper bound $\tilde{\sigma}_{\mathbf{e}}^2(t,i)$ defined in (5.24), while ensuring the distributed consensus on the estimation results among the nodes. Replacing $\sigma_{\mathbf{e}}^2(t,i)$ in (P1) with $\tilde{\sigma}_{\mathbf{e}}^2(t,i)$ yields

$$\begin{aligned} \text{(P2)} \quad \text{minimize} \quad & \lim_{t\to\infty} \max_i \tilde{\sigma}_{\mathbf{e}}^2(i,t) \\ \text{subject to} \quad & \lim_{t\to\infty} |\tilde{\mathbf{s}}(i,t) - \tilde{\mathbf{s}}(j,t)| = 0, \forall i \neq j \end{aligned} \tag{5.25}$$

As can be seen from (5.23) and (5.24), $\tilde{\sigma}_{\mathbf{e}}^2(i,t)$ is related to the matrices $\mathbf{W}$ and $\mathbf{V}$ through

$\mathbf{A}_t$ as $\mathbf{A}_t = \mathbf{A}(i,t) = \mathbf{V}^T\mathbf{W}_i(t) \in \mathcal{R}^{m\times n}$.

Denote $\mathbf{A}_\infty = \lim_{t\to\infty} \mathbf{A}_t$, then (P2) can be solved by identifying $\mathbf{A}_\infty$ that can minimize $\tilde{\sigma}^2_{\mathbf{e}}(i,t)$. We have the following theorem regarding the optimum solution to (P2).

*Theorem 5.2*: The optimum solution to (P2) is $\mathbf{A}^*_\infty = [\mathbf{v}_1^T, \cdots, \mathbf{v}_m^T]^T$, where $\mathbf{v}_i \in \mathcal{R}^n$ is the eigenvector corresponding to the $i$-th largest eigenvalue of the matrix $(\mathbf{R_{xx}}+\sigma^2\mathbf{I}_n)^{-1}\mathbf{R}^T_{\mathbf{sx}}\mathbf{R_{sx}}$.

**Proof:** The proof is in Appendix 5.9.5. ■

The result in Theorem 5.2 indicates that if $\mathbf{A}_t$ converges to a matrix containing the $m$ leading eigenvectors of the $(\mathbf{R_{xx}} + \sigma^2\mathbf{I}_n)^{-1}\mathbf{R}^T_{\mathbf{sx}}\mathbf{R_{sx}}$ as $t \to \infty$, then we can minimize the MSE upper bound. In addition, since $\mathbf{A}_t = \mathbf{A}(i,t)$ for the $i$-th node converges to a constant matrix independent of node index $i$, then the estimates at all nodes are the same, that is, all nodes achieve a consensus on the estimation results.

Since $\mathbf{A}_t$ depends on both $\mathbf{W}$ and $\mathbf{V}$, we will design $\mathbf{W}$ and $\mathbf{V}$ separately such that the optimum condition in Theorem 5.2 is satisfied.

### 5.5.1 Design of Edge Weight Matrix W

The edge weight matrix should satisfy two conditions. First, for a stable system, the linear iteration in (5.5) needs to converge, that is, $\lim_{t\to\infty} \mathbf{W}^t = \mathbf{W}_0$, such that $\lim_{t\to\infty} \mathbf{Y}(t) = \mathbf{W}_0\mathbf{Z}_d\mathbf{V}$. Second, to meet the objective of distributed consensus among all nodes in the network, the information available at all nodes should be the same when $t$ is large. Based on the above two conditions, we can write the converged matrix $\mathbf{W}_0$ as

$$\mathbf{W}_0 = \lim_{t\to\infty} \mathbf{W}^t = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \tag{5.26}$$

where $\mathbf{1}_n$ is a length $n$ all-one vector.

The convergence conditions of $\mathbf{W}$ are shown in the following theorem.

*Theorem 5.3*: [24, Theorem 1] Eq. (5.26) holds, if and only if

$$\mathbf{1}_n^T \mathbf{W}^* = \mathbf{1}_n^T \tag{5.27}$$

$$\mathbf{W}^* \mathbf{1}_n = \mathbf{1}_n \tag{5.28}$$

$$\rho(\mathbf{W}^* - \mathbf{1}_n \mathbf{1}_n^T / n) < 1 \tag{5.29}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix.

Based on Theorem 5.3, the optimum $\mathbf{W}^*$ with the fastest convergence rate can be obtained by solving a semidefinite program (SDP) optimization problem as in [24].

### 5.5.2 Design of sample weight matrix V

The optimum design of the sample weight matrix $\mathbf{V}$ is discussed in this subsection.

Based on the convergence of $\mathbf{W}^*$ in (5.26), we have

$$\lim_{t \to \infty} \mathbf{A}_t = \frac{1}{n} \mathbf{V}^T \tag{5.30}$$

Substituting (5.30) into (5.23) yields

$$\begin{aligned} \tilde{\mathbf{C}}_{\mathbf{ss}}(\infty) &:= \lim_{t \to \infty} \tilde{\mathbf{C}}_{\mathbf{ss}}(t) \\ &= \mathbf{R}_{\mathbf{ss}} - \mathbf{R}_{\mathbf{sx}} \mathbf{V} \left[ \mathbf{V}^T (\mathbf{R}_{\mathbf{xx}} + \sigma^2 \mathbf{I}_n) \mathbf{V} \right]^- \mathbf{V}^T \mathbf{R}_{\mathbf{sx}}^T \end{aligned} \tag{5.31}$$

It should be note that the asymptotic error covariance matrix $\tilde{\mathbf{C}}_{\mathbf{ss}}(\infty)$ is independent of $i$ due to the constraint that $\mathbf{W}_0$ has identical rows.

The optimization problem in (P2) can thus be solved by minimizing trace($\tilde{\mathbf{C}}_{\mathbf{ss}}(\infty)$) in (5.31). Comparing (5.31) and Theorem 5.2, it is clear that we can minimize trace($\tilde{\mathbf{C}}_{\mathbf{ss}}(\infty)$) by setting $\mathbf{V}^* = \mathbf{A}^*_\infty$.

*Corollary 5.3*: With $\mathbf{W}^*$ in given Theorem 5.3, the optimum $\mathbf{V}^*$ that can solve (P2) is $\mathbf{V}^* = [\mathbf{v}_1, \cdots, \mathbf{v}_m]$, where $\mathbf{v}_i$ is the eigenvector corresponding to the $i$-th largest eigenvalue of the matrix $(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)^{-1}\mathbf{R}^T_{\mathbf{sx}}\mathbf{R}_{\mathbf{sx}}$. ■

**Comment 4:** With the edge weight matrix $\mathbf{W}^*$ in Theorem 5.3 and sample weight matrix $\mathbf{V}^*$ in Corollary 5.3, we have

$$\lim_{t\to\infty} \mathbf{A}_t = \lim_{t\to\infty} \mathbf{V}\mathbf{W}^*_i(t) = \mathbf{V}^* = \mathbf{A}^*_\infty \tag{5.32}$$

which minimizes the MSE upper bound as in Theorem 5.2. Thus the design of $\mathbf{W}$ and $\mathbf{V}$ presented in this section can achieve the asymptotically optimum results in Theorem 5.2.

**Comment 5:** With the optimum design of $\mathbf{W}^*$ and $\mathbf{V}^*$, when $t$ is large, the system model in (5.9) or (5.12) becomes

$$\mathbf{y}_\infty := \lim_{t\to\infty} \mathbf{y}(i,t) = \mathbf{V}^*\mathbf{z} \tag{5.33}$$

This is equivalent to projecting the $n$-dimensional vector $\mathbf{z}$ into an $m$-dimension subspace, and this projection is performed in a distributed manner through information exchange among neighboring nodes. In addition, all nodes reach distributed consensus by having the

same $\mathbf{y}_\infty$. The projection is optimized in a way such that the $m$-dimension subspace is aligned with the eigendirection containing the most salient information needed for the minimization of the MSE upper bound. The distributed low dimension projection reduces the amount of information needed to be propagated in the network, and improves the convergence speed of the estimation.

The optimum design presented require the knowledge of the network covariance matrices $\mathbf{R_{xx}}$, $\mathbf{R_{sx}}$, and the noise variance $\sigma^2$, which are also required to implement the iterative LMMSE algorithm in Theorem 5.1. The covariance matrices and noise variance can be estimated in a distributed manner by the nodes in a training phase, which is discussed in the next section.

## 5.6 Distributed learning of spatial covariance

In this section, we present a distributed learning algorithm for the joint estimation of the spatial covariance and noise variance at all nodes.

During the distributed learning, we adopt a parametric model, the Matérn kernel, for the spatial covariance [23]. The Matérn kernel provides parametric flexibility to model a wide range of spatial covariances in practical local spatial processes [9]. The Matérn kernel is represented as

$$\kappa_{v,l}(d) = \frac{2^{1-v}}{\Gamma(v)} \left( \frac{\sqrt{2v}d}{l} \right)^v K_v \left( \frac{\sqrt{2v}d}{l} \right) \tag{5.34}$$

where $\Gamma$ is the Gamma function, $K_v$ is the modified Bessel function of the second kind, $d$ is the distance between two points of interest, and $v$ and $l$ are the smooth and range parameters

to be estimated from the data samples, respectively.

It should be noted that the actual spatial covariance might follow a different model other than the Matérn kernel. Due to the flexibility of the Matérn kernel, our simulation results show that we can still obtain accurate estimates of the second-order statistics even when the real model is different from the Matérn kernel.

With the Matérn kernel, the spatial covariance estimation is equivalent to the estimation of the two model parameters $v$ and $l$. Each node can jointly estimate $(v, l)$ and the noise variance $\sigma^2$ by exchanging data samples with its immediate neighbors.

The distributed learning algorithm contains two steps: distributed maximum likelihood parameter estimation, and distributed parameter consensus. Details of the two steps are given as follows.

### 5.6.1 Distributed Maximum Likelihood Parameter Estimation

In the first step, each node obtains maximum likelihood (ML) estimates of the parameters $(v, l, \sigma^2)$ by using data samples from all of its immediate neighbors.

Assume each node obtains $K$ independent samples of the random field at its own location. Denote the $K$ data samples collected by node $i$ as $\left\{z^{(t)}(\mathbf{c}_i)\right\}_{t=1}^{K}$. The $K$ data samples are then shared with their immediate neighbors. As a result, node $i$ have a total $N_i K$ independent data samples from its neighbors, where $N_i = |\mathcal{N}_i|$ is the number of neighbors of node $i$.

Index the neighbors of node $i$ as $i_1, i_2, \cdots, i_{N_i}$, where $i_k \in \mathcal{N}_i$. Define $\mathbf{z}_i^{(t)} = [z^{(t)}(\mathbf{c}_i), z^{(t)}(\mathbf{c}_{i_1}), \cdots, z^{(t)}(\mathbf{c}_{i_{N_i}})]^T \in \mathcal{R}^{N_i+1}$. Then the covariance matrix of $\mathbf{z}_i^{(t)}$ is

$$\mathbf{R}_{\mathbf{z}_i\mathbf{z}_i} = \mathbf{R}_{\mathbf{x}_i\mathbf{x}_i} + \sigma^2 \mathbf{I}_{N_i+1} \tag{5.35}$$

where the $(m, n)$-th element of the cross-covariance between the samples from nodes $j$ and $k$ are

$$r_{jk} = \mathbb{E}[z^{(t)}(\mathbf{c}_j)z^{(t)}(\mathbf{c}_k)] = \begin{cases} \kappa_{v,l}(\|\mathbf{c}_j - \mathbf{c}_k\|), & j \neq k \\ 1 + \sigma^2, & j = k \end{cases} \tag{5.36}$$

Define the vector $\mathbf{z}_{jk}^{(t)} = [z^{(t)}(\mathbf{c}_j), z^{(t)}(\mathbf{c}_k)] \in \mathcal{R}^2$. Then the covariance matrix is

$$\mathbf{R}_{jk} = \begin{bmatrix} 1 & r_{jk} \\ r_{jk} & 1 \end{bmatrix} + \sigma^2 \mathbf{I}_2 \tag{5.37}$$

Under the assumption that the data is a spatial Gaussian process, the likelihood functions of the unknown parameters can then be formulated as

$$\begin{aligned} p(\mathbf{z}_{jk}^1, \cdots, \mathbf{z}_{jk}^K | v, l, \sigma^2) &= \frac{1}{[2\pi \det(\mathbf{R}_{jk})]^K} \times \\ \exp\left( -\sum_{t=1}^{K} {\mathbf{z}_{jk}^{(t)}}^T \mathbf{R}_{jk}^{-1} \mathbf{z}_{jk}^{(t)} \right), &\quad j, k \in \{i\} \bigcup \mathcal{N}_i \end{aligned} \tag{5.38}$$

The parameters $(v, l, \sigma^2)$ can then be estimated by using ML estimation as

$$(\hat{v}_i, \hat{l}_i, \hat{\sigma}_i^2) = \underset{v,l,\sigma^2}{\operatorname{argmax}} \prod_{j,k \in \{i\} \bigcup \mathcal{N}_i} p(\mathbf{z}_{jk}^1, \cdots, \mathbf{z}_{jk}^K | v, l, \sigma^2) \tag{5.39}$$

The product in the above equation is performed over $\binom{N_i+1}{2}$ node pairs. The ML estimation in (5.39) does not have a closed-form solution, and can be solved via numerical methods such as interior point or grid search.

### 5.6.2 Distributed Parameter Consensus

Once each node obtains their respective estimates on the parameters, they exchange information with their neighbors such that all nodes in the network achieve a global consensus on the estimates of $(v, l, \sigma^2)$. The distributed parameter consensus can be iteratively achieved by using the edge weight matrix $\mathbf{W}^*$ defined in Theorem 5.3.

The training phase is divided into time slots. At the $i$-th slot of the training phase, denote $\hat{\theta}_i(t)$ as the estimated parameter at node $i$, where $\theta \in \{v, l, \sigma^2\}$. We set $\hat{\theta}_i(0) = \hat{\theta}_i$, which is the ML estimate obtained from (5.39). Then the iterative distributed consensus can be carried out in a similar manner as the iterative information propagation described in (5.4)

$$\hat{\theta}_i(t+1) = w_{ii}\hat{\theta}_i(t) + \sum_{j \in \mathcal{N}_i} w_{ij}\hat{\theta}_j(t), i = 1, 2, \ldots, n, \tag{5.40}$$

or in matrix format

$$\hat{\boldsymbol{\theta}}(t+1) = \mathbf{W}^*\hat{\boldsymbol{\theta}}(t) = (\mathbf{W}^*)^t\hat{\boldsymbol{\theta}}(0) \tag{5.41}$$

where $\hat{\boldsymbol{\theta}}(t) = [\hat{\theta}_1(t), \hat{\theta}_2(t), \cdots, \hat{\theta}_n(t)]^T$.

Based on Theorem 5.3, we have

$$\lim_{t \to \infty} \hat{\boldsymbol{\theta}}(t) = \frac{1}{n}\mathbf{1}\mathbf{1}^T\hat{\boldsymbol{\theta}}(0) \tag{5.42}$$

which can also be written as

$$\lim_{t\to\infty} \hat{\theta}_i(t) = \frac{1}{n}\sum_{k=1}^{n} \hat{\theta}_k \tag{5.43}$$

for $\theta \in \{v, l, \sigma^2\}$. That is, every node obtains an average of the estimated parameters from all nodes in the metwork. Even though the analytical results require $t \to \infty$, our numerical results indicate that the process usually converges with the number of iterations on the same order as the number of nodes in the network.

Once all nodes reach a consensus on the estimation results, they can then use the estimated parameters to formulate the spatial covariance matrices following the Matérn kernel. We will show through simulations that the proposed distributed learning algorithm can obtain very accurate estimates of the second-order statistics.

## 5.7 Simulation and experimental results

Simulation and experimental examples are presented in this section to demonstrate the performance of the proposed distributed estimation algorithms in decentralized sensor networks.

### 5.7.1 Simulation Results

In the simulation, we apply the distributed algorithms over synthesized data generated by using the covariance function with power-law kernel, i.e.

$$r_{xx'} = \mathbb{E}[x(\mathbf{c})x(\mathbf{c}')] = \rho^{\|\mathbf{c}-\mathbf{c}'\|} \tag{5.44}$$

where $\rho \in [0, 1]$ is a spatial correlation coefficient and $\|\cdot\|$ measures distance between points in the field. It should be noted that the power-law kernel is different from the Matérn kernel used in the parameter estimation during the training phase described in Section 5.6. We will show through simulation results that the proposed algorithms work well even with mismatch models.

The data are generated by following a two-dimensional Gaussian process with zero-mean and covariance function given in (5.44). The noise is zero-mean Gaussian distributed white uncorrelated samples. The sensor nodes are randomly deployed in an area of size $15 \times 15$ and two nodes are connected by an edge if their distance is less than certain threshold. We can adjust the number of neighbors of each node by adjusting the threshold.

#### 5.7.1.1 Impacts of Imperfect Second-Order Statistics

We first study the impacts of imperfect second-order statistics on the performance of the proposed distributed iterative LMMSE algorithm. Specifically, we compare in Fig. 5.1 the performance between two systems, one with perfect knowledge of the second order statistics, including $\mathbf{R_{xx}}$ and $\mathbf{R_{sx}}$ and $\sigma^2$, the other one with the above parameters estimated with the distributed learning algorithm described in Section 5.6 during the training phase. Even though the Matérn kernel is used in the training phase, the data are generated by following the power-law kernel.

In the simulation, we set $\rho = 0.80$ and the signal-to-noise ratio (SNR) at 15 dB. There are $n = 50$ nodes in the network, and the average number of neighbors each node has is $\frac{1}{n}\sum_{i=1}^{n} N_i = 3.72$. The size of the state vector is $m = 35$. The system tries to estimate $\mathbf{s} = \mathbf{x}$. In the training phase, each node shares with its neighbors $K$ independent data samples, with

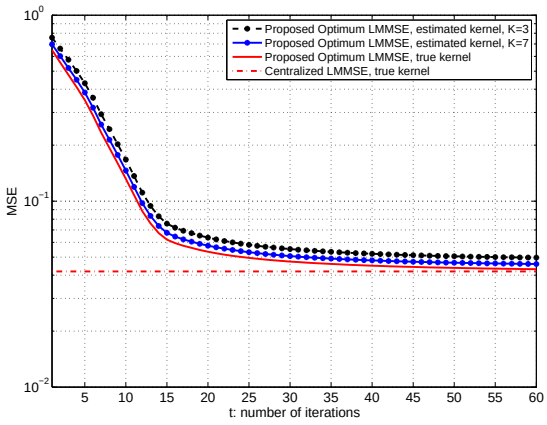



Figure 5.1: Comparison between systems with true and estimated covariances.

$K = 3$ and 7 considered in this example, and all nodes achieve distributed consensus of the estimated parameters in 150 iterations. The simulated MSE of $\hat{\mathbf{s}}$ are calculated by averaging the results from 50 Monte Carlo trials, and they are shown as a function of the number of iterations during the estimation phase.

As can be seen from Fig. 5.1, the performance difference between systems with true and estimated parameters is very small throughout all iterations. At $t = 60$, the MSE for systems with estimated kernels with $K = 3$ and $K = 7$ are $5.0 \times 10^{-2}$ and $4.5 \times 10^{-2}$, respectively, and that for system with true kernel is $4.3 \times 10^{-2}$. Thus the performance loss due to estimated kernel is only 4.6% at $K = 7$. The results indicate that the distributed learning algorithm can achieve very accurate estimation of the second-order statistics, even when there is a mismatch between the actual and training models.

#### 5.7.1.2 Comparison with D-LMMSE

Next we compare the performance between the proposed algorithm with D-LMMSE [18]. The D-LMMSE algorithm requires covariance information $\mathbf{r}_{x_j} := \mathbb{E}[x(\mathbf{c}_j)\mathbf{x}^T]$, $\mathbf{r}_{x_j\mathbf{s}} := \mathbb{E}[x(\mathbf{c}_j)\mathbf{s}^T]$

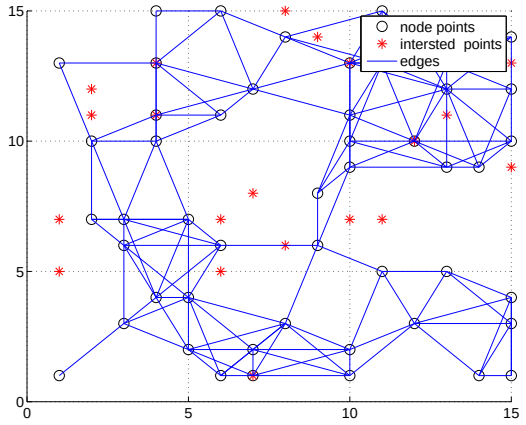



Figure 5.2: The graph of the network with 50 nodes and 110 edges. There are 20 points of interest in the 2-D squared field.

and $\mathbf{R}_{\mathbf{ss}}$ at sensor $j$, while the proposed algorithm requires the covariance information of the entire network. On the other hand, the proposed algorithm requires less information exchange during each iteration. In the proposed algorithm, each node transmits $m \leq \min(n, k)$ values; in the D-LMMSE algorithm, each iteration contains two phases, and each node transmits $n+k$ values in the first phase then each bridge node transmits $n+k$ values in the second phase. The D-LMMSE algorithm requires a set of parameters $c_j$ and $d_j$ at sensor $j$ to be set, and they are numerically optimized to 65 to achieve the fastest convergence.

The sensors are deployed in an area of size $15 \times 15$ as shown in Fig. 5.2. We set $\rho = 0.85$ and SNR $= 20$dB. The network has $n = 50$ nodes (marked as 'o') and 110 edges. The diameter of the network is 9. The sensors will estimate the values at $k = 20$ randomly generated points of interest (marked as '*'). This graph is randomly generated following [24]: 50 nodes are randomly generated, uniformly distributed on the square, then two nodes are connected by an edge if their distance is less than a threshold.

Fig. 5.3 shows the MSE as a function of the number of iterations $t$. Ideal knowledge of

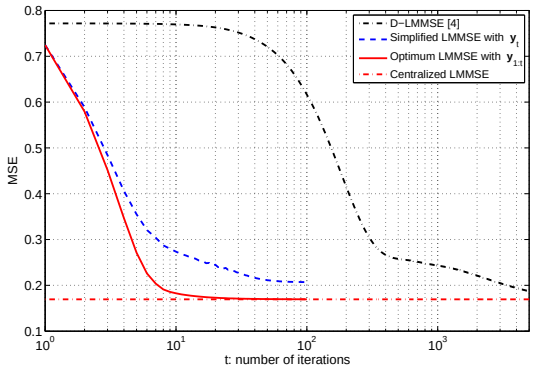



Figure 5.3: Comparison between the proposed algorithm and D-LMMSE [18].

the covariance is assumed for both the proposed algorithm and the D-LMMSE algorithm. The results demonstrate the convergence of the distributed algorithms as time evolves. The number of samples exchanged in each iteration is $m = 10$. For the proposed algorithm, two different types of estimation are performed. The first type is the optimum LMMSE performed by using all previous state vectors $\mathbf{y}_{1:t}$ as in Theorem 5.1. The second type is the simplified LMMSE performed by using the current state vector $\mathbf{y}_t$ as in Corollary 5.2. The MSE for a centralized system with estimation performed at a fusion center is also shown in the figure as reference. As expected, the optimum LMMSE with $\mathbf{y}_{1:t}$ outperforms its low complexity counterpart, but the difference decreases as $t$ increases. The optimum LMMSE converges in $t = 10$ iterations and the simplified LMMSE converges in about $t = 70$ iterations. On the other hand, the D-LMMSE start to converge in about $t = 1,000$ iterations. Even if we consider the cost of the training phase, which requires an additional 150 iterations, the proposed algorithm still converges significantly faster than D-LMMSE.

One of the main factors contributing to the fast convergence of the proposed algorithm is the equivalent low dimension projection achieved with the optimum design of $\mathbf{W}^*$ and $\mathbf{V}^*$ as pointed out in Comment 5. Projecting the $n$-dimension sample vector from all nodes into

a $m$-dimension subspace will significantly reduce the amount of information to be exchanged in the network, thus improve the convergence speed.

#### 5.7.1.3 Comparison with D-RLS

To further demonstrate the performance of the proposed algorithm, we compare its performance with the D-RLS algorithm [3]. To match the terminology used in the D-RLS algorithm with our system model, $\mathbf{y}_t$ is the observation vector, $\mathbf{A}_t$ is the regressor, and $\hat{\mathbf{s}}_t$ is the pre-estimate. We still set $\rho = 0.85$ and SNR = 20dB. The network topology is the same as in Fig. 5.2, except that we will estimate $\mathbf{s} = \mathbf{x}$ in this example, because D-RLS does not exploit spatial covariance and it cannot estimate points that are not sampled by the sensors without the help of an explicit regressor.

The comparison results are shown in Fig. 5.4. Both algorithms converge in about $t = 60$ iterations, but the proposed algorithm achieves significant performance gains over the D-RLS in terms of converged MSE. The MSE of the proposed algorithm converges to that of the centralized algorithm at $10^{-3}$, which is much less than the noise variance. On the other hand, the converged MSE of D-RLS is one magnitude higher at $10^{-2}$, which is the noise variance. The higher MSE floor of D-RLS is partly due to the fact that it does not exploit the spatial covariance among the samples.

It is worth pointing out that the proposed algorithm requires less information exchange than D-RLS. In the proposed algorithm, the only information being exchanged among the neighboring nodes is the state vector $\mathbf{y}_t$. In diffusion RLS, both $\mathbf{y}_t$ (denoted as observations in D-RLS) and $\hat{\mathbf{s}}_t$ (denoted as pre-estimate in D-RLS) are exchanged among the neighboring nodes.

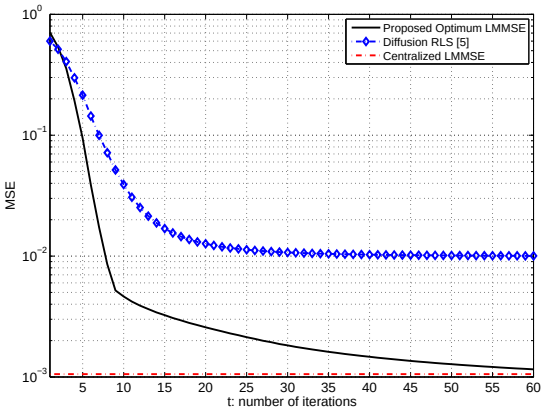



Figure 5.4: Comparison between the proposed algorithm and D-RLS [3].

### 5.7.2 Experimental Results

In the experiment, we apply the proposed algorithm to data collected from an Asian rice field by the department of Electrical Engineering at Shenyang Agricultural University at Shenyang, China. Optical sensors are deployed to measure the Normalized Difference Vegetation Index (NDVI) of the Asian rice at different locations on the field. In this example we use the observations from $n = 11$ locations to estimate the NDVI at $k = 5$ locations. The graph of the network and the relative positions of the points of interest are shown in Fig. 5.5.

The NDVI data are modeled as Gaussian Process with the Matérn covariance. The parameters $(v, l, \sigma^2)$ are estimated with maximum likelihood estimation.

An NDVI value is collected at each location each day for 31 days. On each day, the parameters of the covariance function is estimated by using the data from the previous 10 days with maximum likelihood estimation, and the distributed estimation algorithm is then applied to estimate the NDVI values at the points of interest. The MSE results are averaged over the estimation performed on days 11 to 31.

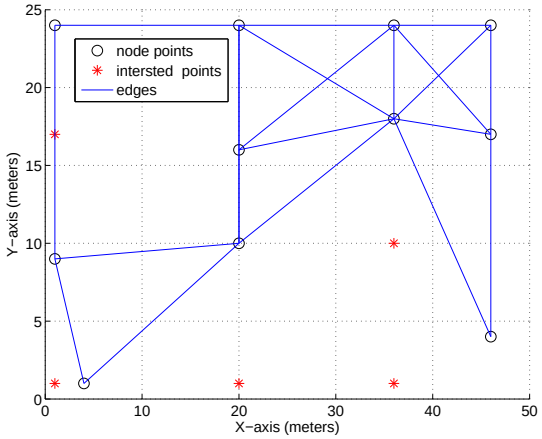



Figure 5.5: Graph of the experiment network and relative positions of the points of interest.

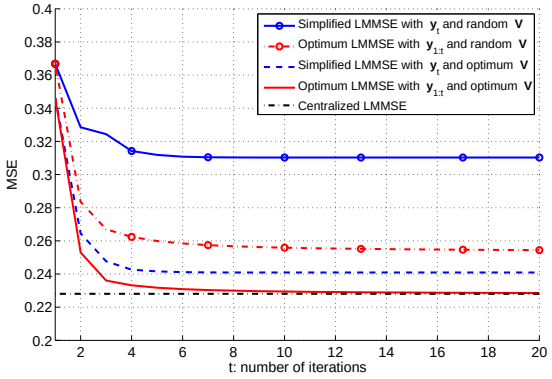



Figure 5.6: MSE as a function of the number of iterations $t$ in experiment.

Fig. 5.6 shows the MSE as a function of the number of iterations $t$. The number of samples exchanged in each iteration is $m = 4$. The MSE obtained in a centralized network with LMMSE performed at a FC is shown as a reference. To better illustrate the convergence of the algorithms, Fig. 5.7 shows the MSE difference between the distributed and centralized algorithm, $\sigma_e^2(t) - \sigma_0^2$, where $\sigma_e^2(t)$ is the MSE of the proposed algorithm, and $\sigma_0^2$ is the MSE at the fusion center in a centralized system.

At any given time slot $t$, the MSE of the proposed system with optimum $\mathbf{V}$ always outperforms that of the system with random $\mathbf{V}$. The MSE of the proposed algorithm converges

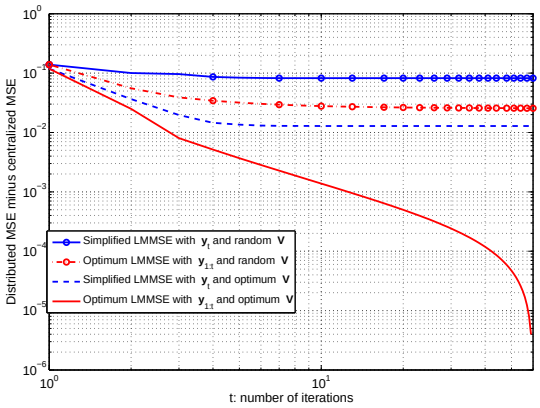



Figure 5.7: Difference between distributed and centralized MSEs as a function of the number of iterations $t$ in experiment.

to that of the centralized algorithm as $t$ increases, and the majority of the performance gain is achieved during the first 5 iterations.

## 5.8 Conclusions

We have studied the distributed estimation of a spatially correlated random field with a decentralized network. Nodes in the network perform estimation of arbitrary points on the random field by iteratively exchanging information with each other. A distributed iterative LMMSE algorithm has been proposed. The key parameters of the algorithm, including the edge weight matrix $\mathbf{W}$ and sample weight matrix $\mathbf{V}$, are designed by following asymptotically optimum conditions when $t \to \infty$. The optimum design equivalently projects the high dimension data collected by all sensors to a low dimension subspace in a distributed manner during information propagation. As a result, the proposed algorithm can significantly reduce the amount of information exchanged in the network, thus improve the convergence speed. Simulation and experimental results have shown that the proposed distributed estimation algorithm outperforms D-LMMSE and D-RLS algorithms in terms of MSE, convergence

speed, and complexity.

## 5.9 Appendices

### 5.9.1 Proof of Lemma 1

The LMMSE estimate of $\mathbf{s}$ can be written as $\hat{\mathbf{s}}_t = \mathbf{U}_t \mathbf{y}_{1:t}$, where $\mathbf{U}_t$ is the MMSE weight matrix. Based on the orthogonality principle $\mathbb{E}[(\mathbf{U}_t \mathbf{y}_{1:t} - \mathbf{s})\mathbf{y}_{1:t}^T] = \mathbf{0}$, we have $\mathbf{U}_t$ satisfying $\mathbf{U}_t \mathbf{R}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}} = \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t}}$.

Denote $\mathbf{p} := [\mathbf{y}_{1:t}^T, \mathbf{s}^T]^T$, and the positive semidefinite matrix $\mathbf{R}_{\mathbf{pp}}$ can be partitioned as $\mathbf{R}_{\mathbf{pp}} = \begin{bmatrix} \mathbf{R}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}} & \mathbf{R}_{\mathbf{y}_{1:t}\mathbf{s}} \\ \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t}} & \mathbf{R}_{\mathbf{s}\mathbf{s}}. \end{bmatrix}$ Then based on [26, Theorem 1.19] we have

$$\mathbf{U}_t = \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t}} \mathbf{R}^{-}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}} \tag{5.45}$$

Combining (5.45) with (5.12) yields (5.13).

### 5.9.2 Proof of Theorem 5.1

First we are going to show that the right-hand sides (RHSs) of (5.13) and (5.16) are equal. We observe that the matrices $\mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t}}$ and $\mathbf{R}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}}$ in the RHS of (5.13) can be written in the form of block matrices as

$$\mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t}} = \begin{bmatrix} \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t-1}} & \mathbf{R}_{\mathbf{s}\mathbf{y}_t} \end{bmatrix}$$

and

$$\mathbf{R}^{-}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}} = \begin{bmatrix} \mathbf{R}_{11} := \mathbf{R}_{\mathbf{y}_{1:t-1}\mathbf{y}_{1:t-1}} & \mathbf{R}_{12} := \mathbf{R}_{\mathbf{y}_{1:t-1}\mathbf{y}_t} \\ \mathbf{R}_{21} := \mathbf{R}_{12}^T & \mathbf{R}_{22} := \mathbf{R}_{\mathbf{y}_t\mathbf{y}_t} \end{bmatrix}^{-} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}.$$

Since $\mathbf{R}_{\mathbf{y}_{1:t}\mathbf{y}_{1:t}}$ is positive semidefinite, according to generalized inversion formula of a block matrix [16, 26], we have

$$\begin{aligned} \mathbf{Q}_{11} &= \mathbf{R}_{11}^{-} + \mathbf{R}_{11}^{-}\mathbf{R}_{12}\mathbf{Q}_{22}\mathbf{R}_{21}\mathbf{R}_{11}^{-} \\ \mathbf{Q}_{22} &= \mathbf{R}_{22}^{-} + \mathbf{R}_{22}^{-}\mathbf{R}_{21}\mathbf{Q}_{11}\mathbf{R}_{12}\mathbf{R}_{22}^{-} \\ \mathbf{Q}_{12} &= -\mathbf{R}_{11}^{-}\mathbf{R}_{12}\mathbf{Q}_{22} \\ \mathbf{Q}_{21} &= \mathbf{Q}_{12}^T \end{aligned}$$

Then (5.13) can be rewritten as

$$\begin{aligned} \hat{\mathbf{s}}(t) =& \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t-1}}\mathbf{Q}_{11}\mathbf{y}_{1:t-1} + \mathbf{R}_{\mathbf{s}\mathbf{y}_t}\mathbf{Q}_{21}\mathbf{y}_{1:t-1} \\ &+ \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t-1}}\mathbf{Q}_{12}\mathbf{y}_t + \mathbf{R}_{\mathbf{s}\mathbf{y}_t}\mathbf{Q}_{22}\mathbf{y}_t \end{aligned} \tag{5.46}$$

On the other hand, based on (5.11), (5.16) can be reformulated as

$$\hat{\mathbf{s}}(t) = \hat{\mathbf{s}}(t-1) + \mathbf{C}_{\mathbf{s}\mathbf{y}_t}(t-1)\mathbf{C}^{-}_{\mathbf{y}_t\mathbf{y}_t}(t-1)(\mathbf{y}_t - \hat{\mathbf{y}}_t(t-1)) \tag{5.47}$$

From (5.14), we have

$$\hat{\mathbf{s}}(t-1) = \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t-1}}\mathbf{R}_{11}^{-}\mathbf{y}_{1:t-1} \tag{5.48}$$

$$\hat{\mathbf{y}}_t(t-1) = \mathbf{R}_{21}\mathbf{R}_{11}^{-}\mathbf{y}_{1:t-1} \tag{5.49}$$

Combining (5.48) and (5.49) into (5.15) yields

$$\mathbf{C}_{\mathbf{s}\mathbf{y}_t}(t-1) = \mathbb{E}[(\hat{\mathbf{s}}(t-1)-\mathbf{s})(\hat{\mathbf{y}}_t(t-1)-\mathbf{y}_t)^T] \tag{5.50}$$

$$= \mathbf{R}_{\mathbf{s}\mathbf{y}_t} - \mathbf{R}_{\mathbf{s}\mathbf{y}_{1:t-1}}\mathbf{R}_{11}^{-}\mathbf{R}_{12} \tag{5.51}$$

Similarly, from (5.15) and (5.49) we have

$$\mathbf{C}_{\mathbf{y}_t\mathbf{y}_t}^{-}(t-1) = (\mathbf{R}_{22} - \mathbf{R}_{21}\mathbf{R}_{11}^{-}\mathbf{R}_{12})^{-} = \mathbf{Q}_{22} \tag{5.52}$$

where the Woodbury matrix identity is used in obtaining the second equality.

Substituting (5.48)–(5.52) into (5.47) and simplifying, we can see that the RHS of (5.47) is the same as that of (5.46).

The proof of (5.17) is similar to the above procedures.

Next we are going to show that $\mathbf{C}_{\mathbf{s}\mathbf{z}}(t)$ can be iteratively calculated with (5.19). Based on (5.11), (5.17) can be alternatively expressed as

$$\hat{\mathbf{z}}(t) = \hat{\mathbf{z}}(t-1) + \mathbf{C}_{\mathbf{z}\mathbf{y}_t}(t-1)\mathbf{C}_{\mathbf{y}_t\mathbf{y}_t}^{-}(t-1)(\mathbf{y}_t - \hat{\mathbf{y}}_t(t-1)) \tag{5.53}$$

Plugging (5.47) and (5.53) into the definition of $\mathbf{C_{sz}}(t)$, we have

$$\begin{aligned}
&\mathbf{C_{sz}}(t) = \mathbb{E}[(\hat{\mathbf{s}}(t) - \mathbf{s})(\hat{\mathbf{z}}(t) - \mathbf{z})^T] \\
&= \mathbb{E}[(\hat{\mathbf{s}}(t-1) + \mathbf{C_{sy_t}}(t-1)\mathbf{C}^-_{\mathbf{y_t y_t}}(t-1)(\mathbf{y}_t - \hat{\mathbf{y}}_t(t-1)) - \mathbf{s}) \\
&(\hat{\mathbf{z}}(t-1) + \mathbf{C_{zy_t}}(t-1)\mathbf{C}^-_{\mathbf{y_t y_t}}(t-1)(\mathbf{y}_t - \hat{\mathbf{y}}_t(t-1)) - \mathbf{z})^T] \\
&= \mathbf{C_{sz}}(t-1) - 2\mathbf{C_{sy_t}}(t-1)\mathbf{C}^-_{\mathbf{y_t y_t}}(t-1)\mathbf{C_{y_t z}}(t-1) \\
&+ \mathbf{C_{sy_t}}(t-1)\mathbf{C_{y_t y_t}}(t-1)^-\mathbf{C_{y_t z}}(t-1) \\
&= \mathbf{C_{sz}}(t-1) - \mathbf{C_{sz}}(t-1)\mathbf{A}_t^T\left(\mathbf{A}_t\mathbf{C_{zz}}(t-1)\mathbf{A}_t^T\right)^-\mathbf{A}_t\mathbf{C_{zz}}(t-1)
\end{aligned}$$

The proofs of (5.18) and (5.20) are similar.

### 5.9.3 Proof of Corollary 5.1

When $t \geq d$, the initial state vector from any node in the network has reached all other nodes in the network through the iterative information exchange described in (5.7). As a result, all elements of the matrix $\mathbf{W}^t$ are non-zero. Therefore the $n \times n$ diagonal matrix $\mathbf{W}_i(t)$, which contains all elements on the $i$-th row of $\mathbf{W}^t$, has full rank $n$, when $t \geq d$.

Based on the assumption that $\mathbf{V}$ is of full column rank, then $\mathbf{A}_t = \mathbf{V}^T\mathbf{W}_i(t)$ is of full rank when $t \geq d$. As a result,

$$\mathbf{A}_t^T\mathbf{B}_t^-\mathbf{A}_t = \mathbf{C}^-_{zz}(t-1), \quad \forall t \geq d \tag{5.54}$$

Substituting the above result into (5.18) yields $\mathbf{C}_{zz}(t) = 0$, for $t \geq d$. From (5.19), we have $\mathbf{C}_{sz}(t) = 0$, for $t \geq d$. Thus from (5.16), it is clear that $\hat{\mathbf{s}}(t) = \hat{\mathbf{s}}(d)$, for all $t \geq d$.

The above analysis is true for all nodes in the network, thus all nodes have the same estimates.

### 5.9.4 Proof of Corollary 5.2

Eqn. (5.22) can be directly obtained from (5.13) in Lemma 1 by replacing $\mathbf{A}_{1:t}$ and $\mathbf{y}_{1:t}$ with $\mathbf{A}_t$ and $\mathbf{y}_t$, respectively. Eqn. (5.23) can be obtained by using the orthogonality principle $\mathbb{E}[(\tilde{\mathbf{s}}_t - \mathbf{s})\mathbf{y}_t^T] = \mathbf{0}$.

Since $\tilde{\sigma}^2_{\mathbf{e}}(t,i)$ is obtained by using $\mathbf{y}_t$ yet $\sigma^2_{\mathbf{e}}(t,i)$ is obtained by using $\mathbf{y}_{1:t}$, it is straightforward that $\tilde{\sigma}^2_{\mathbf{e}}(t,i) \geq \sigma^2_{\mathbf{e}}(t,i)$.

### 5.9.5 Proof of Theorem 5.2

The proof of Theorem 5.2 requires the following lemma.

*Lemma 2*: Consider positive semi-definite (PSD) matrix $\mathbf{A} \in \mathcal{R}^{n\times n}$ , positive definite (PD) matrix $\mathbf{B} \in \mathcal{R}^{n\times n}$, and $\mathbf{X} \in \mathcal{R}^{n\times m}$ with $m \leq n$. Define $\mathbf{Y} = (\mathbf{X}^T\mathbf{B}\mathbf{X})^-\mathbf{X}^T\mathbf{A}\mathbf{X}$. Then we have

$$\text{trace}(\mathbf{Y}) \leq \sum_{i=1}^{m} \lambda_i \tag{5.55}$$

where $\lambda_i$ is the $i$-th largest eigenvalue of $\mathbf{B}^{-1}\mathbf{A}$ or generalized eigenvalue of $(\mathbf{A}, \mathbf{B})$ with associated eigenvector $\mathbf{u}_i$. The equality holds when $\mathbf{X} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$.

**Proof:** The eigenvalue decomposition of $\mathbf{B}^{-1}\mathbf{A}$ is $\mathbf{B}^{-1}\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^{-1}$ where $\mathbf{\Sigma}_n = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ and $\mathbf{U}_n = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$.

In what follows we are going to show that the eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$, $\{\lambda_i\}_{i=1}^n$ are real and non-negative. Since $\mathbf{B}$ is PD, performing Cholesky decomposition of $\mathbf{B}^{-1}$ yields $\mathbf{B}^{-1} = \mathbf{Q}^T\mathbf{Q}$. Thus $\mathbf{I}_n = \mathbf{B}\mathbf{Q}^T\mathbf{Q}$ and

$$\mathbf{I}_n = \mathbf{Q}\mathbf{I}_n\mathbf{Q}^{-1} = \mathbf{Q}(\mathbf{B}\mathbf{Q}^T\mathbf{Q})\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{B}\mathbf{Q}^T \tag{5.56}$$

Denote $\boldsymbol{\phi}_i = (\mathbf{Q}^T)^{-1}\mathbf{u}_i$ , then we have $\mathbf{u}_i = \mathbf{Q}^T\boldsymbol{\phi}_i$. Left multiplying $\mathbf{Q}$ on both side of $\mathbf{A}\mathbf{u}_i = \lambda_i\mathbf{B}\mathbf{u}_i$, we obtain that

$$\mathbf{Q}\mathbf{A}\mathbf{Q}^T\boldsymbol{\phi}_i = \lambda_i\mathbf{Q}\mathbf{B}\mathbf{Q}^T\boldsymbol{\phi}_i = \lambda_i\boldsymbol{\phi}_i$$

which means $\lambda_i$ are also the eigenvalues of the PSD matrix $\mathbf{Q}\mathbf{A}\mathbf{Q}^T$, thus $\lambda_i \geq 0$.

Denote $\boldsymbol{\Phi} = [\bar{\boldsymbol{\phi}}_1, \bar{\boldsymbol{\phi}}_2, \ldots, \bar{\boldsymbol{\phi}}_n]$ as the normalized eigenvector matrix, where $\bar{\boldsymbol{\phi}}_i = \boldsymbol{\phi}_i/\sqrt{\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i}$. Since $\mathbf{Q}\mathbf{A}\mathbf{Q}^T$ is symmetric, based on the spectral theorem, $\boldsymbol{\Phi}$ is an orthonormal matrix, that is, it satisfies $\boldsymbol{\Phi}^T = \boldsymbol{\Phi}^{-1}$. Thus we have $\mathbf{Q}\mathbf{A}\mathbf{Q}^T = \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T$.

Let $\mathbf{V} = \mathbf{Q}^{-1}\boldsymbol{\Phi}$, then we can obtain $\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T$ and $\mathbf{B} = (\mathbf{Q}^T\mathbf{Q})^{-1} = \mathbf{V}\mathbf{V}^T$. Thus we have

$$\begin{aligned}\text{trace}(\mathbf{Y}) &= \text{trace}((\mathbf{X}^T\mathbf{V}\mathbf{V}^T\mathbf{X})^{-}\mathbf{X}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{X}) \\ &= \text{trace}(\mathbf{P}\boldsymbol{\Sigma})\end{aligned}$$

where $\mathbf{P} := \mathbf{V}^T\mathbf{X}(\mathbf{X}^T\mathbf{V}\mathbf{V}^T\mathbf{X})^{-}\mathbf{X}^T\mathbf{V}$ is a projection matrix. Denote $l = \text{rank}(\mathbf{X}^T\mathbf{V}) \leq m$. Since $\mathbf{P}$ is a projection matrix, it has $l$ eigenvalues with value 1 and and $n-l$ zero eigenvalues.

Finally by applying Von Neumann's trace inequality [10] we have

$$\text{trace}(\mathbf{P}\boldsymbol{\Sigma}) \leq \sum_{i=1}^{l} \lambda_i \leq \sum_{i=1}^{m} \lambda_i$$

The equality holds when $\mathbf{X} = \mathbf{U}_m$ because $\mathbf{A}\mathbf{U}_m = \mathbf{B}\mathbf{U}_m\boldsymbol{\Sigma}_m$ then $\mathbf{U}_m^T\mathbf{A}\mathbf{U}_m = \mathbf{U}_m^T\mathbf{B}\mathbf{U}_m\boldsymbol{\Sigma}_m$ and

$$\mathbf{Y} = (\mathbf{U}_m^T\mathbf{B}\mathbf{U}_m)^{-1}\mathbf{U}_m^T\mathbf{A}\mathbf{U}_m = \boldsymbol{\Sigma}_m.$$

■

Now we are ready to prove Theorem 5.2.

From (5.23) and (5.24), solving (P2) is equivalent to maximize the following cost function

$$J(\mathbf{X}) = \text{trace}\Big\{\mathbf{R}_{\mathbf{sx}}\mathbf{X}\left[\mathbf{X}^T(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)\mathbf{X}\right]^{-}\mathbf{X}^T\mathbf{R}_{\mathbf{sx}}^T\Big\} \tag{5.57}$$

where $\mathbf{X} = \mathbf{A}_\infty^T$.

Let $\mathbf{R}_{\mathbf{sx}}^T\mathbf{R}_{\mathbf{sx}} = \mathbf{A}$ and $\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n = \mathbf{B}$. Then the cost function in (5.57) can be rewritten as

$$J(\mathbf{X}) = \text{trace}\Big\{\left[\mathbf{X}^T(\mathbf{R}_{\mathbf{xx}} + \sigma^2\mathbf{I}_n)\mathbf{X}\right]^{-}\mathbf{X}^T\mathbf{R}_{\mathbf{ss}}^T\mathbf{R}_{\mathbf{sx}}\mathbf{X}\Big\} \tag{5.58}$$

$$= \text{trace}\left\{(\mathbf{X}^T\mathbf{B}\mathbf{X})^{-}\mathbf{X}^T\mathbf{A}\mathbf{X}\right\} \tag{5.59}$$

From Lemma 2, we have

$$J(\mathbf{X}) \leq \sum_{i=1}^{m} \lambda_i \tag{5.60}$$

where $\lambda_i$ is the $i$-th largest eigenvalue of $\mathbf{B}^{-1}\mathbf{A} = (\mathbf{R_{xx}}+\sigma^2\mathbf{I}_n)^{-}\mathbf{R}_{\mathbf{sx}}^T\mathbf{R_{sx}}$. The maximum value of $J(\mathbf{X})$ is achieved when $\mathbf{X}$ is formed by the $m$ leading eigenvectors of $(\mathbf{R_{xx}}+\sigma^2\mathbf{I}_n)^{-1}\mathbf{R}_{\mathbf{sx}}^T\mathbf{R_{sx}}$.

## 5.10 References


[1] Sergio Barbarossa and Gesualdo Scutari. Bio-inspired sensor network design. *IEEE Signal Processing Magazine*, 24(3):26–35, 2007.

[2] Alexander Bertrand and Marc Moonen. Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks. *IEEE Transactions on Signal Processing*, 59(5):2320–2330, 2011.

[3] Federico S Cattivelli, Cassio G Lopes, and Ali H Sayed. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, 56(5):1865–1877, 2008.

[4] Jorge Cortés. Distributed Kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827, 2009.

[5] Alexandros G Dimakis, Soummya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[6] Soummya Kar and Jose MF Moura. Consensus+ innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Processing Magazine*, 30(3):99–109, 2013.

[7] Usman A Khan, Soummya Kar, Ali Jadbabaie, and José MF Moura. On connectivity, observability, and stability in distributed estimation. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 6639–6644. IEEE, 2010.

[8] Lin Li, Anna Scaglione, and Jonathan H Manton. Distributed principal subspace estimation in wireless sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):725–738, 2011.

[9] Budiman Minasny and Alex B McBratney. The Matérn function as a general model for soil variograms. *Geoderma*, 128(3):192–207, 2005.

[10] Leon Mirsky. A trace inequality of John von Neumann. *Monatshefte für Mathematik*, 79(4):303–306, 1975.

[11] Esha D Nerurkar, Stergios I Roumeliotis, and Agostino Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1402–1409. IEEE, 2009.

[12] Reza Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.

[13] Reza Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5492–5498. IEEE, 2007.

[14] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[15] Shinkyu Park and Nuno C Martins. An augmented observer for the distributed estimation problem for lti systems. In *American Control Conference (ACC), 2012*, pages 6775–6780. IEEE, 2012.

[16] Charles A Rohde. Generalized inverses of partitioned matrices. *Journal of the Society for Industrial and Applied Mathematics*, 13(4):1033–1035, 1965.

[17] Ali H Sayed and Cassio G Lopes. Adaptive processing over distributed networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 90(8):1504–1510, 2007.

[18] Ioannis D Schizas, Georgios B Giannakis, Stergios I Roumeliotis, and Alejandro Ribeiro. Consensus in ad hoc WSNs with noisy links - Part II: Distributed estimation and smoothing of random signals. *IEEE Transactions on Signal Processing*, 56(4):1650–1666, 2008.

[19] Ioannis D Schizas, Gonzalo Mateos, and Georgios B Giannakis. Distributed LMS for consensus-based in-network adaptive processing. *IEEE Transactions on Signal Processing*, 57(6):2365–2382, 2009.

[20] Ioannis D Schizas, Alejandro Ribeiro, and Georgios B Giannakis. Consensus in ad hoc WSNs with noisy links - part I: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350–364, 2008.

[21] Yunlong Wang and Petar M Djurić. Distributed Bayesian estimation of linear models with unknown observation covariances. *IEEE Transactions on Signal Processing*, 64(8):1962–1971, 2016.

[22] Zuoen Wang, Jingxian Wu, and Jing Yang. Distributed estimation of a spatially correlated random field in decentralized sensor networks. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[23] Zuoen Wang, Jing Yang, and Jingxian Wu. Level set estimation of spatial-temporally correlated random fields with active sparse sensing. *IEEE Transactions on Aerospace and Electronic Systems*, 53(2):862–876, 2017.

[24] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[25] Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70. IEEE, 2005.

[26] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

# Chapter 6

# Conclusions

This chapter summarizes the main contributions of this dissertation and lists some possible directions for the future research.

## 6.1 Contributions

The contents presented in this dissertation focused on the development of the theories and practices of energy aware sparse sensing schemes for the monitoring of random fields that are correlated in the space and/or time domains. The main contributions are summarized as follows. and the main contributions are summarized as follows.

At first we studied energy efficient LSE of random fields correlated in time and space under a total power constraint. We considered uniform sampling schemes of a sensing system with a single sensor and a linear sensor network with sensors distributed uniformly in a line where sensors employ a fixed sampling rate to minimize the LSE error probability in the long term. The exact analytical cost functions and their respective upper bounds of these sampling schemes are developed by using an optimum thresholding-based LSE algorithm. The design parameters of both sampling schemes are optimized by minimizing their respective cost functions. With the analytical results, we identified the optimum sampling period and/or node distance that can minimize the LSE error probability. Analytical and simulation results demonstrate that these sampling schemes can significantly reduce the amount of

data collected by the system while obtain accurate LSE under a stringent power constraint.

Secondly we proposed active sparse sensing schemes with LSE of a spatial-temporally correlated random field by using a limited number of spatially distributed sensors. In these schemes a central controller is designed to dynamically select a limited number of sensing locations according to the information revealed from past measurements, with the objective to minimize the expected level set estimation error. The expected estimation error probability is explicitly expressed as a function of the selected sensing locations, and the results are used to formulate the optimal sensing location selection problem as a combinatorial problem. Two low complexity greedy algorithms were developed by using analytical upper bounds of the expected estimation error probability. Both simulation and experiment results demonstrate that the greedy algorithms can achieve significant performance gains over baseline passive sensing algorithms and the GP Upper Confidence Bound (GP-UCB) level set estimation algorithm.

Lastly we investigated the distributed estimations of a spatially correlated random field with decentralized WSNs. we proposed a distributed iterative estimation algorithm that defines the procedures for both information propagation and local estimation in each iteration. The key parameters of the algorithm, including an edge weight matrix and a sample weight matrix, are designed by following the asymptotically optimum criteria. It is shown that the asymptotically optimum performance can be achieved by distributively projecting the measurement samples into a subspace related to the covariance matrices of data and noise samples. Simulation and experimental results show that all nodes in a large network can obtain accurate estimation results with much less iterations than existing algorithms.

Inspired by the contents what we have done so far, some other topics are presented in

this proposal for the future work.

## 6.2 Future Work

We list two possible directions for future work.

Firstly inspired by the idea of distributed estimation using linear iterations in Chapter 5, we are considering the problem of distributed computation in finite time in a distributed WSN. How to design edge weight matrix and the sample weight introduced in Chapter 5 such that distributed computation can be reached in finite iterations close to the network diameter is of great practical values. In many consensus algorithms [1,6–8,11,12] distributed consensus is reached under various graph typologies asymptotically over an infinite-time horizon. Some study on discrete-time distributed finite-time consensus algorithms has been found in recent literature [2–5, 9, 10, 13] due to their nature of applicability, none of which has addressed the problem of reaching distributed consensus in the discrete time steps equal to the network diameter. We obtained some preliminary results to solve this problem. Each node in the network is required to compute some function of initial values at all nodes through information exchange step by step between their direct neighbors. Each node maintains a state vector initialized by multiple weighted copies of its initial value and updates it by linear iterations using linear combination of its previous state vector and those of all its neighbors. A linear equation system combining the initial value vector and local observation vector from aggregated state vectors at each node can be established. The linear equation system can be solved thus function value of the initial vector can be calculated by running the linear iteration for a finite number of time-steps until the system matrix is full-rank. We can adjust the size of the state vector, which is upper bounded by the size of the network so that the

number of time-steps needed to achieve distributed function calculation can be as small as its lower bound, the diameter of the given graph. We want to investigate further conditions and constraints on this solution that we have.

Secondly, we want to extend our work in Chapter 4 to LSE with mobile path planning. The algorithms and results in Chapter 4 are developed without considering the mobility constraints when a fixed number of samples are taken at each time slot if mobile sensing systems is considered and the analytical performance results can serve as lower bounds for systems with mobility constraints. Specifically we are interested in applying our LSE algorithms into agricultural applications in which drones or other flying robots with camera are used to gather light-spectrum related physical quantities such as the normalized difference vegetation index (NDVI) in massive agricultural fields. Our goal is to estimate the level sets of these physical quantities based on the a limited number of samples taken by drones flying in certain heights over too large field such that it is costly to cover the entire field. Given a fixed flying time, there are more samples taken in higher level which would potentially improve the estimation accuracy; on the other side the samples taken in higher level would suffer more distortion due to path loss and other environmental factors based on the nature of light propagation in air. It is very attractive and practical to balance this tradeoff and design optimum flying heights and paths.

## 6.3 References


[1] Ming Cao, A Stephen Morse, and Brian DO Anderson. Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.

[2] Julien M Hendrickx, Raphaël M Jungers, Alexander Olshevsky, and Guillaume Vankeerberghen. Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, 50(2):635–640, 2014.

[3] Julien M Hendrickx, Guodong Shi, and Karl H Johansson. Finite-time consensus using stochastic matrices with positive diagonals. *IEEE Transactions on Automatic Control*, 60(4):1070–1073, 2015.

[4] Alain Y Kibangou. Finite-time average consensus based protocol for distributed estimation over awgn channels. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5595–5600. IEEE, 2011.

[5] Chih-Kai Ko and Xiaojie Gao. On matrix factorization and finite-time average-consensus. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5798–5803. IEEE, 2009.

[6] Angelia Nedic, Alex Olshevsky, Asuman Ozdaglar, and John N Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[7] Alex Olshevsky and John N Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.

[8] Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5):655–661, 2005.

[9] Guodong Shi and Karl Henrik Johansson. Convergence of distributed averaging and maximizing algorithms part i: Time-dependent graphs. In *American Control Conference (ACC), 2013*, pages 6096–6101. IEEE, 2013.

[10] Shreyas Sundaram and Christoforos N Hadjicostis. Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, 26(4), 2008.

[11] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

[12] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[13] Ye Yuan, G-B Stan, Ling Shi, Mauricio Barahona, and Jorge Goncalves. Decentralised minimum-time consensus. *Automatica*, 49(5):1227–1235, 2013.

# Appendix

## Vitae

Zuoen Wang received the B.S. and M.S. degrees in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2010 and Shenzhen Graduate School of Harbin Institute of Technology, Shenzhen, China, in 2013, respectively. He is currently a Ph.D. candidate in the Department of Electrical Engineering, University of Arkansas, Fayetteville,Arkansas, USA. He is a student member of IEEE. His current research interests include wireless communications, wireless sensor networks and digital signal processing.