UNIVERSITY OF CALIFORNIA, SAN DIEGO

DEPLOYMENT AND ORGANIZATION STRATEGIES FOR SAMPLING-INTERPOLATION SENSOR NETWORKS

A dissertation submitted in partial satisfaction of the requirements for the degree

Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Periklis G. Liaskovitis

Committee in charge:

Professor Curt Schurgers, Chair
Professor Rene L. Cruz
Professor William S. Hodgkiss
Professor Kenneth Kreutz-Delgado
Professor Geoffrey M. Voelker

The Dissertation of Periklis G. Liaskovitis is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2009

DEDICATION

To my family

EPIGRAPH

…a mode of truth, not of truth coherent and central, but angular and splintered.

De Quincey, *Writings*, XI, 68
by way of Jorge Luis Borges, *Evaristo Carriego*

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

This work was supported, in part, by the NSF ECS0622005 SCHURGERS 28647A grant and the UK GORT OF GB 20073248 grant.

Chapter 2 appears, in part, in the *Proceedings of the Third IEEE International Conference on Distributed Computing in Sensor Systems*, *DCOSS 2007*. It is also, in part, under review for publication in the Springer *Journal of Signal Processing Systems*.

Chapter 3 appears, in part, in the *Proceedings of the Second IEEE International Conference on Distributed Computing in Sensor Systems*, *DCOSS 2006*, in the *Proceedings of the Third IEEE International Conference on Distributed Computing in Sensor Systems*, *DCOSS 2007*, as well as in the *Proceedings of the IEEE Wireless Communications and Networking Conference, 2009*. It is also, in part, under review for publication in the Springer *Journal of Signal Processing Systems* and in ACM *Transactions on Sensor Networks*.

Chapter 4 is, in full, under review for publication in the *Proceedings of the IEEE Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, *ISSNIP 2009*.

In all aforementioned cases, I was the primary researcher and Curt Schurgers supervised the research.

In the realm of actual living people, my first and foremost thanks go to my advisor, Professor Curt Schurgers. I wish I knew this guy more. Because, from what I do know, he seems like someone who knows how to enjoy life. And this has a tremendously positive impact on his general attitude, his perspective towards research

and the guidance he has given me over the years. It was he who told me that the PhD is about solving interesting toy problems and maybe learning something in the process. In the end, I can sincerely say it came down to this. I warmly thank him for helping me during my timid first steps -when I was right off the boat so to speak- by simply urging me to think. In technical matters, I took example from his astonishing ability to do away with dead-end problems and impractical details right from the start (fortunately for me, he cannot exercise this ability here). He taught me how to see the big picture and this stands for a lot in the reality of engineering. In tactical matters, he taught me good judgment, persistence in the face of (almost) impossible deadlines and how it is important not to touch my itchy nose when presenting in front of an audience. And all this, without imposing on my weekends. I guess here is the place to put in writing that, although I did not know him before I came here, he is the best advisor I could have wished for.

I also want to thank the other members of my committee. Although our interaction was not very frequent, they have appreciated and greatly enhanced my work in their own fashion. I thank Professor Ken Kreutz-Delgado for his illustrious personality and delivery of the Parameter Estimation course. These were my inspiration for beginning to explore the mathematics underlying signal processing and, ultimately, a strong influence on my own research work. I thank Professor Rene Cruz for instigating me with diligence on Poisson processes as well as for his unique approach to introductory probability: I have not to date seen a more intuitive way than starting from coin flips. I thank Professor Bill Hodgkiss for introducing me to the vast relevant literature of the marine people, the ones who can actually spend quarters of

research at Baja California or Micronesia. Last, but not least, I want to thank Professor Geoff Voelker for his sharp and insightful feedback, remarkably and solely harvested from minute presentations.

One of the most important tips and tricks I learnt early on for progressing through the levels of the PhD game, was to pay attention to my peers. Peers were indeed a perennial source of inspiration and motivation for me. They are too many to explicitly mention here: the fellow graduate students at the WISL lab who have put up with me as a creature of the lab, but have also encouraged me, the Greek gang of engineering, who taught me to thrive in chaos of my own creation, and the surrounding –usually female- people who were making chaos tolerable and sometimes warm. I consider all these people to be my lifelong friends. And still, what would the PhD be without all the extracurricular activities? I am eternally thankful to the engineers-rock-climbers association of UCSD and particularly to the Greek chapter, of which I was a proud vice-president for some twenty months. They taught me perseverance in the face of adversity. And to the argentine tango and aikido people of San Diego. They taught me poetry in the face of adversity.

I should not omit to express my gratitude to Raymond Chandler and his spiritual child, detective Philip Marlowe. A gritty, cynical man who is downright honest, yet enough of an altruist to take action far beyond what he wants us all to think of him. Had it not been for a small passage in the Long Goodbye, where Marlowe is heading south to Tijuana from his base in Los Angeles, perhaps my interest in this small part of the world would never have been sparked. San Diego was a harbor of lights. And along with that, the ocean, the coastline, Sunset Boulevard, Santa Monica, San

Francisco, the culmination of the Californian pop culture of the 60s and 70s, really shaped things for me. The true essence of *light* and of *noir*, the tao of existence, will be forever present. Raymond, I hope you are enjoying yourself, wherever you are.

On a sidenote, I would like to thank Honore de Balzac, Henri Beyle Stendhal, Friedrich Nietzsche, Fyodor Dostoyevsky, Robert Louis Stevenson, Gilbert Keith Chesterton, Jorge Louis Borges, Herman Hesse, Kurt Vonnegut, Hunter Stockton Thompson, Tibor Fischer, Chuck Palahniuk, Raymond Smullyan, Jiddu Krisnamurti, Manu Bazzano, Mark Twight, Kazuo Koike, Goseki Kojima, Yamamoto Tsunetomo, Alan Moore, Kostis Palamas, Thanassis Veggos, Jason Bourne, Jack Sparrow, Rorschach, Corto Maltese and Heath Ledger's Joker. They have changed things for me. Forever.

Finally, I thank my parents and my brother. I owe everything to them.

What does this all have to do with a PhD in Electrical Engineering? I do not know. The forces that drive us henceforth are not always easily explainable. The black belt lesson is that you always learn; of the vastness of things you don't know. This therefore was neither a speech at the onset of a career, nor at the finishing line of one. It was merely a note on *becoming*. Sorry, if I have been misleading at times. I didn't do it on purpose. I almost never do.

VITA AND PUBLICATIONS

VITA

2002	Diploma in Electrical and Computer Engineering, National Technical University of Athens
2003	Research Assistant in Department of Informatics and Telecommunications, University of Athens
2003	Recipient of the University of California, San Diego, Departmental Fellowship.
2006	Master of Science in Electrical Engineering, University of California, San Diego
2007	Candidate of Philosophy in Electrical Engineering, University of California, San Diego
2009	Teaching Assistant in Electrical and Computer Engineering Department, University of California San Diego
2009	Doctor of Philosophy in Electrical Engineering, University of California, San Diego

PUBLICATIONS

1. **Liaskovitis, P.** and Schurgers, C., "Efficient Sensing Topology Management for Spatial Monitoring with Sensor Networks", submitted to Springer Journal of Signal Processing Systems, 2009.

2. **Liaskovitis, P.** and Schurgers, C., "Leveraging Redundancy in Sampling-Interpolation Applications for Sensor Networks: A Spectral Approach", submitted to ACM Transactions on Sensor Networks, 2007.

3. **Liaskovitis, P.** and Schurgers, C., "Energy Consumption of Multi-hop Wireless Networks under Throughput Constraints and Range Scaling", accepted for publication in ACM Mobile Computing and Communications Review, 2007.

4. **Liaskovitis, P.** and Schurgers, C., "Designing Sensor Networks for Spatial Interpolation", submitted to ISSNIP, June 2009.

5. **Liaskovitis, P.** and Schurgers, C., "Energy Efficient Min-Max Spatial Monitoring with Wireless Sensor Networks", in WCNC, April 2009.

6. **Liaskovitis, P.** and Schurgers, C., "Leveraging Redundancy in Sampling-Interpolation Applications for Sensor Networks", in DCOSS, June 2007.

7. **Liaskovitis, P.** and Schurgers, C., "A Distortion-Aware Scheduling Approach for Wireless Sensor Networks", in DCOSS, June 2006.

ABSTRACT OF THE DISSERTATION

DEPLOYMENT AND ORGANIZATION STRATEGIES FOR SAMPLING-INTERPOLATION SENSOR NETWORKS

by

Periklis G. Liaskovitis

Doctor of Philosophy in Electrical Engineering (Communication Theory and Systems)

University of California, San Diego, 2009

Professor Curt Schurgers, Chair

Networks of wireless micro-sensors are envisioned to be the prominent choice for on-site monitoring of physical locations. A wide range of practical applications has been conceived and studied in recent years for this engineering regime: habitat and wildlife monitoring, smart buildings and disaster response are only a few representative examples. However, there are also unique challenges faced by the sensor network paradigm: energy resources for individual sensors are limited. Efficient approaches are necessary to ensure prolonged autonomous operation of the network, while still providing quality of service to the user application at all times.

Here, we focus on situations where the wireless sensor network functions as a distributed sampling system and sensors periodically sample a physical phenomenon of interest, e.g. temperature. Samples are then used to construct a spatially continuous estimate of the phenomenon through interpolation, over time.

We examine two distinct classes of practical sampling-interpolation scenarios. In the first one we are given a large ensemble of sensors which have already been deployed. The goal is then to reactively devise a maximum number of disjoint subsets of sensors, such that data from each of them can individually support the desired interpolation accuracy. Energy efficiency is achieved by reducing the amount of data packets communicated across the network. In the second one we have to proactively manage deployment of the network from scratch. The objective is then to use a minimum number of sensors so as to again support the desired interpolation accuracy. Cost effectiveness is achieved here by using a smaller network to begin with.

To tackle the challenges of these scenarios we utilize the Hilbert space of second order random variables and define interpolation quality on the basis of Mean Squared Error (MSE). Times series of values measured at individual sensors can provide finite dimensional approximations of these random variables and facilitate algebraic manipulations within the Hilbert space framework. The associated covariance matrix succinctly captures sensor correlations and enables novel solutions to the aforementioned problems. Through extensive simulations on synthetic and real sensor network data our proposed solutions are shown to possess strong advantages compared to other approaches.

CHAPTER 1

1. INTRODUCTION

1.1 The Vision for Wireless Sensor Networks

For more than a decade now, researchers have been weaving the vision of the wireless sensor network (WSN): a new paradigm for connecting humans to the physical world. In this vision, sensors are small, untethered devices that can sense diverse physical phenomena, including but not limited to, mechanical forces, chemical concentrations and electromagnetic fields. Deployed in areas of interest, they will gather, process and disseminate information according to application needs [Cul04], [Rag02].

The focal point in this evolution is not the technological sophistication of individual nodes themselves, but the properties of their collection as a whole. Energy storage capabilities of devices are limited and energy efficiency emerges as a goal of utmost importance for prolonged autonomous operation of the network. In return, the sensor network effectively provides a collective alternative to traditional remote monitoring techniques. Instead of employing few very expensive sensors, e.g., satellites, to monitor areas of interest from afar, a big group of sensing devices will be embedded directly onto these areas, enjoying physical proximity to the phenomena or events to be monitored. Depending on their cost, devices may exploit multi-modal sensing capabilities ranging from full visual motion capture to simple temperature or

pressure readings. Most importantly, they will be wirelessly networked together as a transparent ‘sensing entity’ able to furnish a much richer, real-time representation of the surrounding physical environment than remote monitoring.

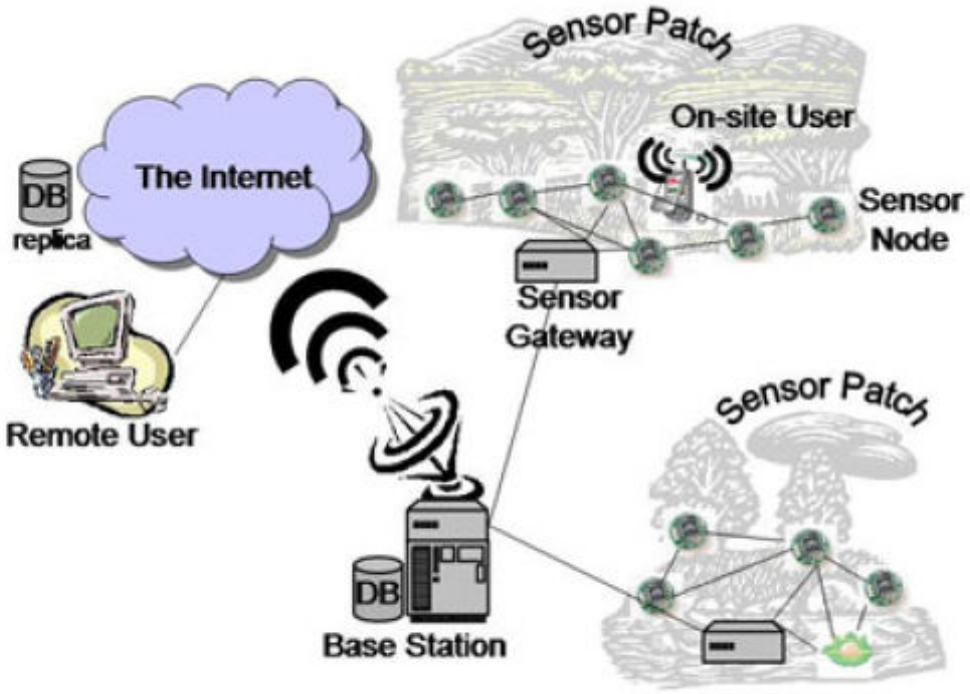

Figure 1.1: Great Duck Island deployment

The spectrum that can potentially utilize this sensing entity is very multifaceted: environmental and structural monitoring, wildlife tracking, disaster response and smart physical environments are only a few of the available scenarios. There are numerous examples of such WSNs deployed within the last decade. Two of the most well known ones are the Great Duck Island deployment [Mai02], [Sze04a], [Sze04b] realized through collaboration between the University of California Berkeley and Intel Research, as well as the ExScal deployments [Aro05], [Bap05], [ExS04] led by a team

of Ohio State University researchers. The former aimed at habitat monitoring for a species of sea birds, while the objective of the latter was detection and classification of multiple intruder types over an extended perimeter. A sketch of the tiered network architecture used to extract information from the Great Duck Island deployment is shown in Figure 1.1. The ExScal project on the other hand, actually involved the largest sensor network deployments to date, consisting of approximately 10000 nodes. More research motivated deployments have been realized and we will give brief accounts of them in later sections.

On a commercial scale, according to recent industry studies [Con05], the market for WSNs is growing and is predicted to continue to grow rapidly in the next years, as their potential is getting more widely understood and more relevant opportunities emerge. A curve depicting the expected growth of sensors deployed is shown in Figure 1.2:

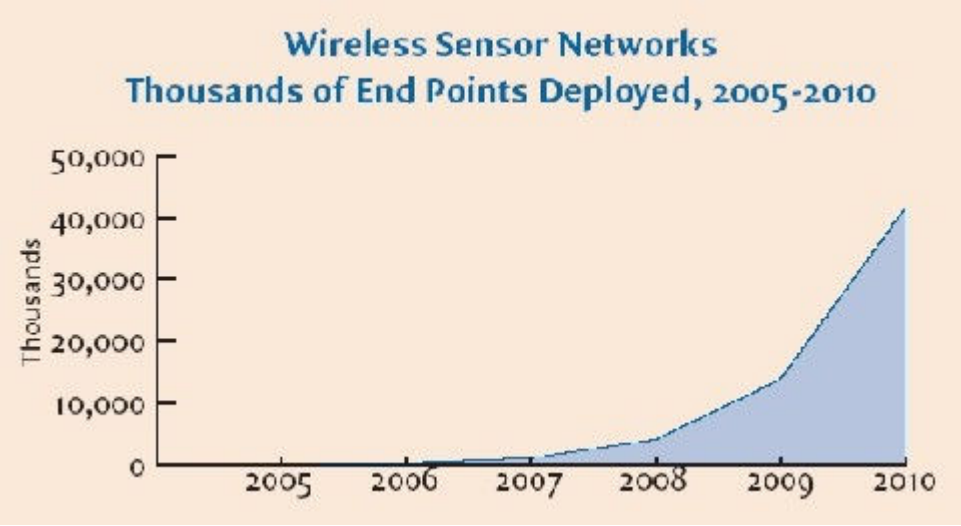

Figure 1.2: Anticipated number of sensor nodes deployed

However, the sheer spectrum of envisioned applications also prevents the existence of solutions that will be universal to all WSN related considerations. No energy efficiency or robustness strategy for example can be blindly applicable to all situations. *Classes* of solutions should instead target *classes* of applications.

1.2 A Taxonomy of Sensor Network Applications

The work we present here necessarily focuses on a specific subset of WSN applications. To make our application domain clearer, we will contrast it to others and pinpoint its distinct characteristics. Prevalent taxonomies of sensor network applications characterize them on the basis of data delivery requirements, or on the basis of actual objectives [Li08]. Here we adopt the latter criterion, which essentially views applications under the prism of two distinct classes.

In *event detection* applications, the network should be able to provide a good indication of where and when an 'event' occurs within its area. Consider the example of a network of sensors that can measure and report temperature to an external data sink. An 'event' could be said to occur when the temperature readings of sensors in a specific area rise above a certain threshold, thus indicating a fire. The objective would then be 'detect a fire'. Another event detection objective would be intruder detection and classification. Immediate real world examples facilitating such an application are the ExScal deployments mentioned in the previous section.

A subclass of event detection applications are *target tracking* applications. Their goal is to first detect an event and then follow its movement through the physical area of the network. Note that target tracking in this context assumes that there is no sensor

physically attached to the target. For the temperature measuring scenario for example, the goal would be to follow the movement of a fire. Potential real world examples of target tracking applications can again be found in the ExScal deployments, where sensors can be configured to track the movement of specific intruders within the network.

On the other hand, our work focuses on *continuous monitoring* applications, where each sensor periodically produces data that must be conveyed to the sink. The objective is to reliably compute a function on collected data. This is in direct contrast to identifying outliers, which is the case for event detection. The most prevalent among such objectives is *interpolation* of sensor measurements. The network as a whole acts as a distributed sampling system for the value of a physical quantity of interest. Samples obtained are interpolated to produce a continuous estimate for this quantity in time and space. In the temperature measuring scenario for example, sensors sample temperature over time and at any given point in time these samples can be interpolated to determine how temperature varies in space, i.e., even in locations where no sensors exist. Monitoring environmental phenomena over a specific area is an important example of this kind of applications. The physical quantities of interest can include ambient phenomena, such as temperature, humidity and light intensity as well as more localized ones, such as rain precipitation levels, wind velocity and water contamination.

Current deployments relevant to interpolation applications are the Networked InforMechanical System (NIMS) sensor, which has been successfully deployed to gather chemical measurements in the San Joaquin and Merced rivers in California

[Har06], as well as networks for indoor temperature monitoring nurtured by Intel Research [Gue05]. The Sensorscope project in EPFL [Sen07] has also demonstrated moderately sized deployments of up to 125 solar powered sensing stations, each measuring key environmental data such as air temperature and humidity, surface temperature, incoming solar radiation, wind speed and direction, precipitation, soil water content, and soil water suction. The stations have been deployed over the years in various formations at the EPFL campus or at surrounding sites. Additional examples include networks for habitat monitoring such as the already discussed Great Duck Island deployment and PODS [Bia02].

Query-driven applications can be considered as a subclass of continuous monitoring applications. In this case, rather than each sensor periodically reporting its measurements, the sink could query a specific subset of sensors for their measurements. This enables the sink to extract information at a different resolution or granularity, from different regions in space and in an on-demand fashion. Sensor networks serving query-driven applications often possess the additional characteristic of consisting of sensors embedded on mobile agents, e.g., animals. Real world examples of such applications are wildlife and plant monitoring projects such as Zebranet [Jua02], Shared Wireless Infostation Model (SWIM) [Sma03] and Hogthrob [Hog07].

We anticipate that both major classes of applications, i.e., event detection and continuous spatiotemporal sampling will be of importance as the sensor network vision grows. However, each one of these classes has a conceptually distinct purpose and therefore demands different tools to study effectively. Event detection needs every

specific point in the observation field to be ‘covered’ by a sensor, because any such point could potentially be the origin of an event. Proximity of sensors to events is modeled by such notions as sensing range (see Chapter 2). Interpolation on the other hand does not value specific points, but, in a sense, whole areas of the field in as much as they contribute to an accurate spatial estimate. Tools radically different from sensing range are needed here to characterize sensing quality. In this dissertation we will focus only on interpolation applications.

1.3 Why is Spatial Interpolation Important?

With the term ‘spatial interpolation’ we will hereafter refer to the operation of interpolating samples of a physical quantity gathered at a *fixed* time instant. Considering for simplicity a 2-dimensional sensor network, spatial interpolation effectively results in the construction of a continuous surface of values also referred to as ‘spatial profile’. We focus on spatial interpolation, because it plays a vital role in a lot of envisioned or even realized sensor network scenarios. In addition to earlier examples of such scenarios, a few more are given below:

- Precision agriculture: Spatially varying application of nutrients and fertilizers [Int04] is often necessary to refine crop production. This can greatly benefit from knowledge of how humidity levels and soil synthesis vary over space.
- Habitat monitoring: An important goal of disciplines such as zoology and botanology is the development of reliable models that can predict behavior of various plant and animal species. Such a development needs accurate

information about the values of numerous environmental parameters (e.g., humidity, atmospheric pressure, temperature, infrared radiation, total solar radiation and photosynthetically active radiation [Bia02], [Mai02]) not just at specific sites, but over entire areas of interest.

- Monitoring of freshwater quality: To restore and maintain quality of water supplies in human dominated regions [Har06] the levels of various hydrologic, chemical and ecological parameters need to be taken into account. Since water deposits often stretch over large areas, sampling and interpolation with sensor networks can greatly contribute to the characterization of such quantities.
- Intelligent buildings: Buildings of the future are envisioned to considerably improve the comfort level of inhabitants by constantly monitoring indoor environment characteristics such as temperature, humidity and airflow and acting accordingly, e.g. through precise ventilation and air conditioning. Interpolation of such quantities through a carefully tuned WSN emerges as a cost-effective solution to achieve this monitoring-actuating functionality.
- Urban planning: On an even scale larger than that of a building, studying the concentrations of contaminants (e.g. carbon oxides) in the atmosphere can enable better land-use decisions.

Spatial interpolation essentially provides a link between measuring a quantity at specific locations and then inferring how this quantity varies over whole areas of interest. It is a prominent task within various scenarios and has justifiably attracted increased attention from the research community in recent years.

1.4 Problem Definition

To date, there has been research work addressing various systemic aspects of WSNs that perform continuous monitoring. Issues such as Medium Access Control, routing, localization and others have been explored [Kri05]. However, there has been little work on the fundamental question of how many sensors are needed and where exactly, in order to support the task of spatial interpolation. The problem we are examining in this dissertation can be summarized as:

Manage how many sensors are to be sensing and producing data at any given time and where they should be located, so that construction of spatial profiles for a physical quantity is possible at a given level of accuracy.

Our question therefore is one of managing sampling points for interpolation. Answering this question is directly related to the notions of longevity and cost-effectiveness for the network. For example, if, out of many sensors deployed over an observation field, only a few are deemed necessary to provide an accurate enough interpolation, the rest have the option of not sensing and not sending data through the network at all. These redundant sensors can be themselves rotated into the sensing functionality if needed. This will increase the lifetime of the network as a whole not only because fewer packets are injected and communicated through it but also because 'backup' sensors are available to take over the sensing task should the initial ones deplete their energy reserves. In a different scenario, we may be able to flesh out the important locations to perform sensing at, before actually deploying the network. If

sensors are then deployed only at such locations, the whole deployment will be smaller and therefore more cost effective.

1.5 Reactive vs. Proactive management (or which sensors we need vs. where we need them)

Our purpose in this dissertation is to tackle the problem of efficient sampling and interpolation with WSNs, relying on as few modeling assumptions as possible. In comparison to relevant work, this means that we do not assume prior knowledge about the spatiotemporal behavior of the physical phenomenon of interest. Such prior knowledge for example could be information about the joint distribution of the data or of the second order statistics of an underlying spatiotemporal random process or even that measured signals are compressible in some known basis. Thought in another way, regardless of the interpolation model used to describe the data, we do not assume that we know anything about the *specifics* of the model prior to the deployment. Furthermore, the only assumption we do make is knowledge of an absolute upper bound N_{max} on the number of sensors necessary to provide accurate enough interpolation. This means, that if we deploy N_{max} sensors completely at random, the interpolation achieved will surely satisfy some application and model specific criterion. Knowledge of such an upper bound is not limiting because, by definition, it can be very loose.

Given this basis, the first class of scenarios we will examine is those where we are allowed exactly one ‘shot’ at deploying the network. This essentially means that all sensors are deployed simultaneously (or almost simultaneously) in a single pass,

remain where they were initially deployed and no addition of sensors is possible after that. Given this setup, a sensible choice is to initially deploy the maximum number of sensors N_{max}, i.e., to initially *overdeploy*. This creates *sensing redundancy*. Our goal would then be to decide *which sensors will be sensing and sending data over time* to achieve accurate enough interpolation. Because any technique thus applied is a *reaction* to the already existing network, we will call this task *reactive management* of the network. As will elaborate in Chapter 3 the challenge for us here is to devise multiple subsets of sensors, each individually adequate for accurate enough interpolation and schedule operation of the network accordingly.

A practical scenario amenable to reactive management emerges when the region to be monitored is largely inaccessible. For instance, habitat monitoring at natural reserves (e.g. the Great Duck Island) precludes frequent re-entering of the deployment site. In the same vein, monitoring of remote regions or hostile territory is likely to be served by randomly scattering sensors from an airplane at one shot [Don04], [Sun05], [Zha03]. Another relevant example can be found at situations where the cost of devices is so small as to permit unplanned deployments. Future scenarios for WSNs, envision thousands of tiny, cheap electromechanical sensors embedded in the interior paint of a building [Cul04]. After the network has been deployed (i.e. the building painted) reactive management strategies are needed to ensure quality of interpolation in face of the great number of devices and the randomness of deployment.

The second class of scenarios we will examine is those where we have the option of deploying the network in multiple passes. This means that we are allowed to deploy sensors, have them gather data and re-enter the observation field at will. It is then our

goal to decide on *where to deploy a minimum number of sensors*, so that accurate enough interpolation based on them is possible. In this case, all decisions are made concurrently with the deployment itself and we call this task *proactive management* of the deployment. As we will elaborate in Chapter 4, for such scenarios, we follow a two step approach: the first step is to efficiently 'explore', i.e., estimate the statistics of the underlying random process by *incrementally* placing sensors at appropriate locations and the second step is to select deployment locations facilitating spatial interpolation. The challenge is of course to use as few sensors as possible altogether.

Proactive management can be applied to situations where the cost of individual sensors is so large as to prohibit randomly scattering a large number of them and tighter control over the deployment is actually possible. A salient example would be deploying sophisticated meteorological sensors, such as those of EPFL [Sen07], to perform spatial interpolation. In general, any small experimental deployment targeting spatial interpolation, such as the ones currently operating in various research institutions, would fall under the proactive management paradigm.

Both reactive and proactive management can be seen under a unifying light: the notion of the *exact minimum* number of sensors necessary for accurate enough interpolation, according to the given criterion. Essentially this represents the best possible deployment in terms of number of sensors, given a particular interpolation model. Calling this number N_0, the aim of our different strategies is schematically shown at Figure 1.3. Reactive management effectively tries to select subsets with size close to N_0. Proactive management on the other hand, tries to estimate the statistics of

the phenomenon well enough to be able to reliably predict the whole deployment corresponding to N_0.

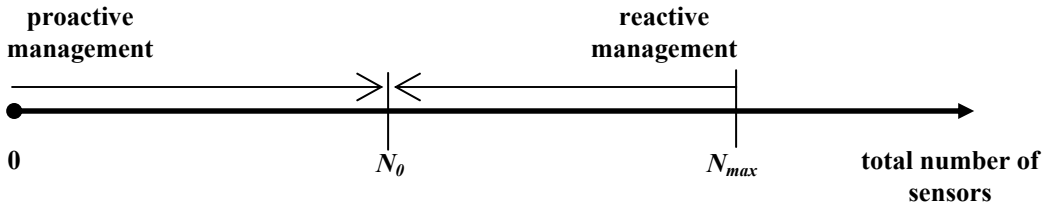

Figure 1.3: Schematic representation of reactive and proactive management for deploying a WSN

1.6 Dissertation Outline

To study interpolation in this dissertation we use the notion of second order statistics for an underlying spatiotemporal random process. Details on these ideas are provided in Chapter 2; an integrated mathematical framework is presented there, able to characterize the quality of interpolation contingent upon the deployment and the monitored physical phenomenon. Chapters 3 and 4 examine reactive and proactive management of the network respectively. Chapter 5 then discusses open problems and future perspectives. These include interpolation model generalizations, network related issues, as well as the more general problem of multi-pass deployments. An account of related work is given at the end of each individual chapter.

1.7 Contributions

An important aspect of our contribution relates to the fact that existing work in the area makes assumptions on the form of the statistics governing the underlying

physical phenomenon: the overwhelmingly prevalent assumption is Gaussian statistics. Even then, associated statistical models (e.g. distribution priors and parameter priors) need to be initialized based on some form of expert knowledge, usually originating from past deployments or measurements. In many practical situations however, assumptions on the statistics are not desirable or even pertinent [Jin06] and may lead to inaccurate conclusions. In addition, although past deployments and associated data provide a solid basis for statistical inference, they may not be always be available.

On the other hand, we relinquish the need for such limiting assumptions and instead build on a Hilbert space framework for interpolation. The Hilbert space is that of second order random variables; time series measured by individual sensors can be considered as their finite dimensional approximations. Although properties of this particular Hilbert space are well known, its finite-dimensional interpretation in the context of a sensor network is a first contribution of this dissertation. The notion of Mean Squared Error (MSE) is utilized, aided by very relaxed notions of prior knowledge about the physical phenomenon at hand. Specifically, we only assume mean square ergodicity in time and knowledge of an upper bound on the number of necessary sampling points, i.e., sensors. Correlations among sensors are captured in a covariance matrix and algebraic tools can be applied to this matrix. This has the potential of being a useful tool in sensor network processing even beyond the specific problems we consider in this dissertation. Part of our pursuit has in fact led us to draw parallelisms with the well known sparse approximation problem in signal processing.

Overall, our approach offers a consistent and elegant treatment for the task of interpolation with a WSN.

1.8 Impact

The key impact of our work then lies in that it provides greater flexibility to WSN deployment. The phenomenon to be monitored need not have been characterized a-priori. This effectively means that the network practitioner does not need access to past observational data or have to set up costly expert assessments in preliminary stages of the network lifecycle. Rather he can start gathering measurements in an informed manner very early on and save on precious resources. The essential core of our work is thus that it brings network practitioners on step closer to the reality of actual deployments.

CHAPTER 2

2. HILBERT SPACES FOR INTERPOLATION

2.1 Background

Our goal in this chapter is to analyze the task of spatial interpolation with a wireless sensor network (WSN) in a mathematically rigorous manner. Spatial interpolation is intertwined with the statistical correlation among the readings of different sensors. If, for example, readings gathered at two distinct spatial locations are correlated, then, to obtain a sufficiently accurate interpolation, we may not need to place sensors in both of these locations or, if sensors are already present, we may not need to use samples from both of them. The main theme will be to characterize such correlation and effectively link it to the interpolation operation.

Different application classes for WSNs (see section 1.2) impose fundamentally different notions of correlation. For an event detection application, the objective would be to determine when and where an event occurs, i.e., when and where an outlier definitely exists in the sensor readings. Consider the example of a WSN monitoring temperature inside a warehouse: increased temperature readings may indicate the proximity of a hot object. Any point in space can potentially be the source of an event, so a possible strategy would be to have a minimum number of sensors k sensing each point in space. This strategy is well known as k-coverage [Abr04], [Wan03]. To associate an event with a sensor, it is intuitively appealing to define a *sensing range*,

as the maximum distance from a sensor over which an event can be reliably detected [Car04], [Hua03], [Sli01]. Correlation among sensors is dictated by their proximity to a particular event: sensors are correlated if they are close the same event. Since proximity is defined by sensing range so is the notion of correlation.

On the other hand, for a spatial interpolation application the goal would be to construct a continuous surface in space, i.e., defined even in areas devoid of sensors. In the temperature monitoring scenario, it would be desirable to know for example how temperature varies as one is moving away from the windows or air-conditioning units. To accomplish this, measurements are viewed as spatiotemporal samples indexed by the spatial locations of sensors that produced them. Samples are sent through the network to a data sink and combined by means of an *interpolation scheme* in a non-trivial manner that can no longer be captured by the notion of sensing range.

There are two major factors crucial to all possible interpolation schemes: the distribution of sensor nodes in space and the correlation characteristics of the physical phenomenon to be monitored. The main theme of this chapter is in fact how to analyze the interpolation operation itself in association with these two factors. How is interpolation affected by the existence or not of sensors in a spatial region? How is interpolation quality improved (or diminished) by including (respectively, omitting) sensors that are really close to each other? Furthermore, how do answers to these questions change in accordance to faster or slower variations of the monitored physical phenomenon over space?

In what follows, we describe a framework that allows us to capture correlation in a spatial interpolation context. This will be used in the remainder of the dissertation to devise effective management strategies.

2.2 Network Model

Consider a sensor network which, at some point in time, consists of N sensors, indexed 1 through N, scattered over an observation field F. We will refer to the 2-D positions of sensor p with the tuple $\boldsymbol{X}_0^p = (x^p, y^p)$, p=1…N. Sensor positions are needed for interpolation itself, and assumed to be known with sufficiently high accuracy by running a localization service in the network [Lan03], [Pat04]. At discrete time instants t_i, a subset of the sensors measures the value of a physical quantity of interest (e.g. temperature). Sensors are time synchronized at a coarse level, so that they can be considered to be sampling at roughly the same time t_i.

The physical phenomenon is modeled as the spatial realization of a random process at t_i, denoted by $S(\boldsymbol{x}, t_i)$, where vector $\boldsymbol{x}$ represents 2-D coordinates in the observation field F. An approximation of this realization is constructed at a data sink by interpolating data values reported only by the sensors 'active' at t_i, along with their positions. 'Active' sensors are those that actually sense and communicate their data. Note that non-active sensors are not generating sensing data. They may be in an energy efficient sleep state or forwarding data produced by active modes. The collection of active sensors is hereafter referred to as *subset* or *set*. A subset is represented by the Boolean vector $\boldsymbol{m}_k$, of length N, where each element is (1) for an active sensor and (0) otherwise (by convention, $\boldsymbol{m}_0$ will refer to the set of all available

sensors). The exact positions of the sensors comprising the subset will be denoted by the tuple $\boldsymbol{X}_k^p = (x^p, y^p)$, p=1…$|\boldsymbol{m}_k|$, where $|\cdot|$ is the size of a set.

We consider interpolation that is *linear on values measured* by the sensors. This is an assumption widely employed in existing literature [Dig06], [Gue04], [Ste99], [Zha03] and covers a broad range of interpolation techniques [Mar01], [Moo00]. Denoting the random process values at sensor locations as $S(\boldsymbol{X}_k^p, t_i)$ the interpolation at point $\boldsymbol{x}$ can be generally expressed as:

$$\hat{S}_k(\boldsymbol{x}, t_i) = \lambda_0(\boldsymbol{x}, \boldsymbol{m}_k) + \sum_{p=1}^{|\boldsymbol{m}_k|} \lambda_p(\boldsymbol{x}, \boldsymbol{m}_k) \cdot S(\boldsymbol{X}_{\boldsymbol{k}}^{\boldsymbol{p}}, t_i) \tag{2.1}$$

where $\{\lambda_p(\boldsymbol{x},\boldsymbol{m}_k)\}_{p=0\ldots N}$ are coefficients describing how a specific interpolation scheme depends on the particular subset of sensors $\boldsymbol{m}_k$ $\forall t_i$. The collection of interpolations $\hat{S}_k(\boldsymbol{x}, t_i)$ for $\forall \boldsymbol{x} \in F$ is based only on values originating from sensors in $\boldsymbol{m}_k$ and forms a surface also referred to as *spatial profile*. A single subset $\boldsymbol{m}_k$ can generate surfaces for multiple t_i. The idea is that interpolation is performed with one subset for some time and then control is turned over to another subset. For example, the surfaces $\hat{S}_2(\boldsymbol{x}, t_1)$, $\hat{S}_2(\boldsymbol{x}, t_5)$, $\hat{S}_2(\boldsymbol{x}, t_8)$ are assumed to have been generated by $\boldsymbol{m}_2$, reporting at time instants t_1, t_5, t_8 respectively.

2.3 Hilbert Space Methodology – Fundamentals

To analyze spatial interpolation with a sensor network, we develop a methodology that maps the network onto an equivalent Hilbert space. A Hilbert space

is a collection of elements, indiscriminately referred to as points or vectors, which can be entities of any kind and have appropriate operations defined on them. These operations are addition, scalar multiplication, inner product and norm of an element (for more details see [Deu01], [Moo00]). Hilbert spaces are widely used as a tool in approximation, because results within their framework are amenable to simple geometric interpretations.

A fundamental question arising is what should be considered an element of the space in our sampling-interpolation setting. Existing work in WSN literature that has used the notion of Hilbert spaces (e.g. compressive sensing [Baj06], [Dua06]), has utilized the space $\Re^m$, i.e., the space of finite length vectors of real elements equipped with the usual Euclidean inner product. Each of these vectors typically holds collected measurements of a single sensor. Here, we make an alternate choice of Hilbert space structure that we believe is much more naturally suited to our particular problem: the Hilbert space of random variables with finite second order moments. To the best of our knowledge, ours ([Lia07a]) is the first interpretation of a sensor network as an instance of this particular Hilbert space, which could potentially find use beyond the context examined here.

To be more specific, the measured value of the physical phenomenon $S(\boldsymbol{x}, t_i)$ at location $\boldsymbol{x}$ for a specific, fixed time instant t_i can be viewed as a random variable. The completed span of these random variables, i.e., all their linear combinations and limits of Cauchy sequences thereof form a Hilbert space [Cra67], [Mar01]. We define the inner product, the (induced) norm and the distance between two elements in this space as (* denotes the complex conjugate):

$$< S(\boldsymbol{x}_1,t_1), S(\boldsymbol{x}_2,t_2) >= E[(S(\boldsymbol{x}_1,t_1) - \tilde{\mu}(\boldsymbol{x}_1,t_1)) \cdot (S(\boldsymbol{x}_2,t_2) - \tilde{\mu}(\boldsymbol{x}_2,t_2))^*] \quad (2.2)$$

$$\|S(\boldsymbol{x},t_1)\|^2 =< S(\boldsymbol{x},t_1), S(\boldsymbol{x},t_1) >= E[|\, S(\boldsymbol{x},t_1) - \tilde{\mu}(\boldsymbol{x},t_1)\,|^2] \quad (2.3)$$

$$\|S(\boldsymbol{x_1},t_1) - S(\boldsymbol{x}_2,t_2)\|^2 = E[|\, S(\boldsymbol{x_1},t_1) - S(\boldsymbol{x_2},t_2) - (\tilde{\mu}(\boldsymbol{x_1},t_1) - \tilde{\mu}(\boldsymbol{x_2},t_2))\,|^2] \quad (2.4)$$

where $\tilde{\mu}(\cdot,\ \cdot)$ is a mean function that varies in time and space. The expectation operator $E[\cdot]$ is with respect to the joint probability density function describing the values of the physical phenomenon at any finite number of points in continuous space and at any finite number of time instants, assuming that such a function exists.

Viewing our problem under the light of this particular Hilbert space structure, entails some assumptions on the random process that models the physical phenomenon at hand. These assumptions are in fact very common and have been extensively used to describe real sensor network data [Jin06], [Kra06] as we do here. For completeness of presentation we will formalize them in the remainder of this subsection.

To begin with, the norm of an element as expressed in eq. (2.3) assumes that the random process has finite second order moments (i.e. the process is a *second order process*). Equations (2.2) (2.3) and (2.4) in fact describe standard formulation for the Hilbert space underlying second order processes. This is an assumption overwhelmingly used in existing literature [Kra06], [Per04], [Vur06], [Zha03] with Gaussian processes as the main representative. Our approach is more generic, however, as it targets distortion in the mean square sense, regardless of the underlying probability distribution.

Additional assumptions are wide sense stationarity and mean square ergodicity in the time index. Wide sense stationarity practically means that the spatial characteristics of the process up to second order are time-invariant. Time invariance is a valid assumption in many cases and depends on the nature of the application. In the indoor temperature monitoring scenario for example, the time scale of change for the mean function would likely be in the order of seasons. This time scale of change will be hereafter referred to, when applicable, as the *coherence time of the application*. Mean square ergodicity practically ascertains ability to estimate ensemble averages from time averages. This has been implicitly assumed in the past when manipulating real sensor network data [Kra06]. Formally, the following relations are assumed:

$$E[S(\boldsymbol{x}_1,t)] = \widetilde{\mu}(\boldsymbol{x}_1,t) = \mu(\boldsymbol{x}_1) \tag{2.5}$$

$$< S(\boldsymbol{x}_1,t_1), S(\boldsymbol{x}_2,t_2) >= \widetilde{C}(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1-t_2) \tag{2.6}$$

$$\lim_{T\to\infty}[\frac{1}{2\cdot T}\cdot\int_{-T}^{T}|\, S(\boldsymbol{x}_1,\boldsymbol{x}_2,t)\,|^2 dt] = \widetilde{C}(\boldsymbol{x}_1,\boldsymbol{x}_2,0) \tag{2.7}$$

where $\mu(\cdot)$ is a spatial mean function and $\widetilde{C}(\cdot,\cdot,\cdot)$ is a space-time covariance function. Equations (2.5) and (2.6) impose wide-sense stationarity while eq. (2.7) is the mean square ergodicity condition.

In light of the stationarity and ergodicity assumptions, the spatial mean function represents a spatially varying mean in the data that does not change over time and can be obtained by averaging measurements over time. In the temperature scenario for example, a spatial trend could exist due to certain parts of the observation field being more exposed to direct sunlight than others. The space-time covariance function on the

other hand, quantifies the statistical similarity between readings gathered at different spatial locations and different times.

Our ultimate goal is to obtain spatial interpolations of the physical phenomenon for *specific time instants*. In this case, only a subgroup of all existing vectors and inner products in the Hilbert space needs to be examined, specifically those that correspond to fixed time instants t_i:

$$< S(\boldsymbol{x}_1, t_1), S(\boldsymbol{x}_2, t_1) >= \widetilde{C}(\boldsymbol{x}_1, \boldsymbol{x}_2, 0) = C(\boldsymbol{x}_1, \boldsymbol{x}_2) \tag{2.8}$$

The inner product in eq. (2.8) is quantifying the correlation between readings at two spatial locations for the same time instant. All relevant inner products are collectively described by the function $C(\cdot, \cdot)$ obtained from eq. (2.6) for $t_1 = t_2$, which will be hereafter referred to with the term *spatial covariance function*.

It can be seen from that the spatial covariance function is not dependent on the specific time instant of observation t_i. Furthermore, it can be any function bounded over the observation field, i.e., spatial stationarity is not assumed. This essentially means that the correlation between readings at two points on the field is allowed to depend on their location, i.e., the covariance structure can be non-stationary. For example, at locations closer to windows or heat sources temperature readings may vary more incoherently for proximate sensors than they do at dark or isolated locations, e.g. under desks. Equation (2.8) provides us with a model sufficiently rich to capture these spatial non-stationarities (also referred to as inhomogeneities) which are in fact commonly encountered in real life physical phenomena and corresponding measurements [Kra06].

As a conclusion, the inner product and norm described by eq. (2.2) and (2.3) are of major importance, because they directly incorporate both the sensor locations and the correlation of the monitored physical phenomenon. Furthermore, the advantage of the Hilbert space structure over mere identification of the sensors as correlated random variables is that we get a rich geometry on these random variables. This enables us to manipulate a set of deterministic vectors instead, with all relevant statistical information captured in the inner product.

2.4 Hilbert Space Methodology – Spatial Interpolation

In this section, our goal is to translate spatial interpolation of the physical phenomenon into Hilbert space terms. Specifically, we will show how the interpolation operation can be abstracted solely by means of the inner product of the space.

Assume that we wish to obtain an approximation of the form (2.1) for the value of the physical phenomenon at location $\boldsymbol{x}$, based on a subset $\boldsymbol{m}_k$ within our network of N sensors. The set of random variables $S(\boldsymbol{x}, t_i)$ across the whole observation field, i.e., $\forall \boldsymbol{x} \in F$, for a fixed time instant t_i, comprises a Hilbert space [Cra67]. We will denote this "ambient" Hilbert space as H_S. By virtue of the ergodicity assumptions discussed in the previous section, properties of this space do not change across different time instants. Random variables corresponding to the (time-invariant) locations of sensors are vectors in this Hilbert space, i.e., there is a one to one correspondence between a specific set of sensors and a specific set of vectors. Therefore, in what follows, the

terms ‘vectors’ and ‘sensors’ will be used interchangeably. The subset $\{S(\boldsymbol{X}_k^p, t_i)\}$, p=1…$|\boldsymbol{m}_k|$, for fixed t_i defines a finite dimensional subspace H_{Xk} [Cra67].

The interpolation problem for point $\boldsymbol{x}$ now becomes a problem of approximating the element $S(\boldsymbol{x}, t_i)$ of the Hilbert space H_S with the finite dimensional subspace H_{Xk}. A well known property of Hilbert spaces that optimally addresses this problem is “approximation through orthogonal projection” [Cra67], [Moo00]. Orthogonal projection basically gives an element of the subspace H_{Xk} for which the Mean Squared Error (MSE) of interpolation $\hat{S}_k(\mathbf{x}, t_i)$ is minimized:

$$MSE(\boldsymbol{x}, \boldsymbol{m}_k) = E[(S(\boldsymbol{x}, t_i) - \hat{S}_k(\boldsymbol{x}, t_i))^2] = \left\| S(\boldsymbol{x}, t_i) - \hat{S}_k(\boldsymbol{x}, t_i) \right\|^2 \tag{2.9}$$

The MSE is a commonly used measure to test fidelity of interpolated spatial profiles from sensor network data [Yu04]. Note that, compared to the general norm expression (2.4), the MSE of eq. (2.9) takes into account the fact that the process $S(\boldsymbol{x}, t)$ is real. The optimal linear interpolator of the form (2.1) can now be viewed as a combination of two terms:

$$\hat{S}_{k,opt}(\boldsymbol{x}, t_i) = \underbrace{\lambda_{0,opt}(\boldsymbol{x}, \boldsymbol{m}_k)}_{bias\ of\ H_{Xk}} + \underbrace{\sum_{p=1}^{|\boldsymbol{m}_k|} \lambda_{p,opt}(\boldsymbol{x}, \boldsymbol{m}_k) \cdot S(\boldsymbol{X}_k^p, t_i)}_{orthogonal\ projection\ onto\ H_{Xk}} = \lambda_{0,opt}(\boldsymbol{x}, \boldsymbol{m}_k) + \hat{S}_{k,orth}(\boldsymbol{x}, t_i) \tag{2.10}$$

The linear combination of sensor vectors in the second term corresponds to orthogonal projection onto the subspace H_{Xk}, while the bias term is equal to $E[S(\boldsymbol{x}, t_i) - \hat{S}_{k,orth}(\boldsymbol{x}, t_i)]$, in other words, to the mean of a certain element of the ambient Hilbert space H_S [Ste99]. Since the bias term depends on the actual vector $S(\mathbf{x}, t_i)$, where there is no

sensor, computation of the best linear interpolator (2.10) in practice relies on models for the spatial mean function $E[S(\boldsymbol{x}, t_i)]$ [Dig06], [Ste99].

A particularly useful observation however, is that the *optimal* (i.e., minimum) MSE induced by such an interpolator is actually the error of orthogonal projection onto the subspace H_{Xk}. The first term in eq. (2.10) effectively renders the best linear interpolator unbiased. Assuming that the spatial covariance function $C(\cdot, \cdot)$ of eq. (2.8) is completely known, we can write the following:

$$G_{\boldsymbol{k}} = \begin{bmatrix} C(\boldsymbol{X}_k^1, \boldsymbol{X}_k^1) & \dots & C(\boldsymbol{X}_k^1, \boldsymbol{X}_k^{|\boldsymbol{m}_k|}) \\ \dots & \dots & \dots \\ C(\boldsymbol{X}_k^{|\boldsymbol{m}_k|}, \boldsymbol{X}_k^1) & \dots & C(\boldsymbol{X}_k^{|\boldsymbol{m}_k|}, \boldsymbol{X}_k^{|\boldsymbol{m}_k|}) \end{bmatrix} \qquad \boldsymbol{c}_k(\boldsymbol{x}) = \begin{bmatrix} C(\boldsymbol{x}, \boldsymbol{X}_k^1) \\ \dots \\ C(\boldsymbol{x}, \boldsymbol{X}_k^{|\boldsymbol{m}_k|}) \end{bmatrix} \tag{2.11}$$

The matrix G_k will be hereafter referred to as the Grammian matrix corresponding to the subset $\boldsymbol{m}_k$ and to $c_k(\boldsymbol{x})$ as the covariance vector of point $\boldsymbol{x}$ with the subset $\boldsymbol{m}_k$. Then the MSE of the optimal linear interpolation, i.e., of orthogonal projection, is given according to Hilbert space theory by [Deu01], [Ste99]:

$$MMSE(\boldsymbol{m}_k, \boldsymbol{x}) = \min_{\hat{S}} \{E[(\hat{S}_k(\boldsymbol{x}, t_i) - S(\boldsymbol{x}, t_i))^2]\} = C(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{c}_k^T(\boldsymbol{x}) \cdot G_k^{-1} \cdot \boldsymbol{c}_k(\boldsymbol{x}) \tag{2.12}$$

Where $MMSE(\cdot, \cdot)$ denotes the Minimum MSE. An alterative form for the MMSE is [Deu01]:

$$MMSE(\boldsymbol{m}_k, \boldsymbol{x}) = C(\boldsymbol{x}, \boldsymbol{x}) - \frac{(\sum_{p=1}^{|\boldsymbol{m}_k|} \left| C(\boldsymbol{x}, \boldsymbol{X}_k^p) \right|)^2}{\sum_{p=1}^{|\boldsymbol{m}_k|} \sum_{q=1}^{|\boldsymbol{m}_k|} C(\boldsymbol{x}, \boldsymbol{X}_k^p) \cdot C(\boldsymbol{x}, \boldsymbol{X}_k^q) \cdot C(\boldsymbol{X}_k^p, \boldsymbol{X}_k^q)} \tag{2.13}$$

Our first contribution is thus viewing the MSE of the best possible linear interpolation with a subset of sensors at a point $\boldsymbol{x}$ as the error of orthogonal projection of the vector $S(\boldsymbol{x}, t_i)$ onto this subset of sensors. Furthermore, it can be seen from eq. (2.12) and (2.13) that this error can be computed solely through knowledge of the values of the spatial covariance function $C(\cdot, \cdot)$ at specific location pairs. The orthogonal projection error will serve in what follows as the means of characterizing different subsets of sensors with respect to the quality of spatial interpolations they can provide.

As a final remark, we should point out that eq. (2.12) assumes that the Grammian matrix G_k is invertible. This is equivalent to the approximating subspace H_{Xk} having as many dimensions as the number of vectors that comprise it, i.e., the vectors $\{S(\boldsymbol{X}_k^p, t_i)\}$, p=1…$|\boldsymbol{m}_k|$ must be linearly independent. Equation (2.13) also assumes linear independence and, additionally, that the covariance vector $c_k(\boldsymbol{x})$ is not the all-zero vector. This translates to the sensor at point x being correlated with at least one of the sensors in the subset $\boldsymbol{m}_k$. In the next section we argue that for a very common type of network deployment and for a stationary underlying process, vectors corresponding to sensors are indeed linearly independent on the average so that both eq. (2.12) and (2.13) can be utilized.

2.5 A Property of Poisson Deployments Sampling Stationary Processes

Consider the scenario where sensors are dropped from a mobile agent, e.g. an airplane over an otherwise inaccessible region. It is then very probable that sensors will rely at random positions in 2-D space. Our interest in the Poisson distribution stems from the fact that it is a mathematically attractive method of describing such randomness in a WSN deployment. By definition, a Poisson deployment pattern implies that for a given observation field and a given number of sensors, any one of them is equally probable to rely anywhere on this field, i.e., sensors are uniformly distributed on the field [Ken98]. In this section, our goal is to prove that, for Poisson distributed sensor nodes and a spatially stationary physical phenomenon, the resulting Hilbert space vectors are linearly independent of each other on the average. This essentially means that no single sensor can be perfectly described by a linear combination of other sensors.

We initially consider a one-dimensional sensor network where sensor positions $\boldsymbol{X}_0^p = (x^p, y^p)$, p=1…N constitute an orderly Poisson process of constant rate β on the observation interval [0, L]. Similarly, we consider the monitored phenomenon as a one-dimensional wide sense stationary process *S(x)* with correlation function $C(x)$ and associated power spectral density $\Phi(\omega)$. An element of the ambient Hilbert space H_S spanned by the random variables *S(x)* for all real x can be expressed as:

$$A = \sum_{k=1}^{n} a_k \cdot S(x_k, t_i) \tag{2.14}$$

The norm of an element as well as the inner product and the distance between two elements are given by eq. (2.2)-(2.4).

Consider now the Hilbert space H_Φ of functions in the form:

$$a(\omega) = \sum_{k=1}^{n} a_\kappa \cdot e^{j\omega x_k} \qquad \int_{-\infty}^{+\infty} |a(\omega)|^2 \cdot \phi(\omega) d\omega < \infty \tag{2.15}$$

The associated inner product is:

$$< a(\omega), b(\omega) >_{H_\Phi} = \int_{-\infty}^{+\infty} a(\omega) \cdot b^*(\omega) \cdot \phi(\omega) d\omega = \sum_{k=1}^{n} \sum_{j=1}^{n} a_k \cdot b_j^* \cdot C(x_k - x_j) =< A, B >_{H_S} \tag{2.16}$$

and, similarly as before, the distance between two elements will be:

$$\|a(\omega) - b(\omega)\|^2 = \int_{-\infty}^{+\infty} |\alpha(\omega) - b(\omega)|^2 \cdot \phi(\omega) d\omega = E[|A - B|^2] \tag{2.17}$$

Since inner products and distances in the two spaces coincide, we can define an isometry mapping between the spaces H_S and H_Φ:

$$I[S(x, t_i)] = e^{j\omega x} \qquad x \in R \tag{2.18}$$

The mapping $I(\cdot)$ holds for finite sums of random variables or functions as shown in eq. (2.16) and (2.17) and can be extended by arguments of mean square convergence to the entire spaces [Cra67], [Mar01]. Isometry means that associated spaces can be treated as equivalent for projection or distance problems. In other words, sums of the form (2.14) can be manipulated as sums of the form (2.15). The advantage we gain by considering H_Φ in place of H_S is that elements of the former are deterministic and, in addition, amenable to Fourier analysis tools.

In order to prove linear independence on the average, we first look at linear combinations of vectors in the isometric subspace. We want to prove that no vector of the form $e^{j \cdot \omega \cdot x_n}$ can be written as a combination of other vectors, when averaging

over all realizations $\{x_p\}$, $p=1...N$, i.e., over all Poisson deployments of the given rate. Suppose the opposite is true. Then for a non-trivial choice of the coefficients c_p, the following expression should hold:

$$E[\varepsilon_n] = \sum_{p=1}^{n} c_p \cdot E[e^{j \cdot \omega \cdot x_p}] = 0 \tag{2.19}$$

The expectation term in (2.19) is the characteristic function of x_p, i.e., of the length of the interval from the origin to the p-th point of the process. This length is often called *recurrence time of order p*. For a Poisson point process, its probability density function $f_p(u)$ depends only on p and the rate β through the following closed form [Beu70]:

$$f_p(u) = \frac{\beta \cdot (\beta \cdot u)^{p-1} \cdot e^{-\beta \cdot u}}{(p-1)!}, u \geq 0 \tag{2.20}$$

Plugging the definition of a characteristic function into (2.19) we get equivalently:

$$\sum_{p=1}^{n} c_p \cdot \int_{-\infty}^{\infty} f_p(u) \cdot I_{(0,\infty)}(u) \cdot e^{j \cdot \omega \cdot u} \cdot du = 0 \Leftrightarrow$$

$$\int_{-\infty}^{\infty} (\sum_{p=1}^{n} c_p \cdot f_p(u) \cdot I_{(0,\infty)}(u)) \cdot e^{j \cdot \omega \cdot u} \cdot du = 0 \Leftrightarrow$$

$$\sum_{p=1}^{n} c_p \cdot f_p(u) = 0, \forall u \geq 0 \tag{2.21}$$

where $I_{(0,\infty)}(u)$ denotes the step function and the last equation follows from the uniqueness of the Fourier transform. Equation (2.21) can be further scrutinized as:

$$\sum_{p=1}^{n} c_p \cdot \frac{\beta \cdot (\beta \cdot u)^{p-1} \cdot e^{-\beta \cdot u}}{(p-1)!} = 0, \forall u \geq 0 \Leftrightarrow$$

$$e^{-\beta \cdot u} \cdot \sum_{p=1}^{n} c_p \cdot \frac{\beta \cdot (\beta \cdot u)^{p-1}}{(p-1)!} = 0, \forall u \geq 0 \Leftrightarrow$$

$$\sum_{p=1}^{n} c_p \cdot \frac{\beta^p}{(p-1)!} \cdot u^{p-1} = 0, \forall u \geq 0 \Leftrightarrow$$

$$c_p \cdot \frac{\beta^p}{(p-1)!} = 0, \forall p \Leftrightarrow c_p = 0, \forall p \tag{2.22}$$

where the last equation follows from the linear independence of any subset of monomials, over any interval of the real line, i.e.:

$$\sum_{p=0}^{n} a_p \cdot u^p = 0, \forall u \geq 0 \Leftrightarrow a_p = 0, \forall p \tag{2.23}$$

Equation (2.23) contradicts our initial hypothesis and therefore, on the average, no vector in a Poisson deployment can be a linear combination of other vectors. To generalize for the case of multidimensional scenarios, we observe that a multi-dimensional Poisson sampling process can be degraded into a one dimensional one by keeping only a single (e.g. the first) coordinate of each sensor position; by definition of the Poisson process, for a fixed number of sensors this coordinate will be uniformly distributed in the observation interval and, since involved distributions are continuous, the probability that two of the resulting points will coincide is zero. The corresponding correlation function would then be formulated by taking only differences of the chosen coordinate into account. The resulting proof arguments are the same. ■

We have now proved the following lemma:

Lemma 2.1: For a deployment where the positions of the sensors form a Poisson point process with constant rate β and the monitored random process $S(\mathbf{x})$ is wide sense stationary, vectors $\{S(\mathbf{X}_0^p, t_i)\}_{p=1..N}$ are linearly independent on the average.

Note that the lemma indicates what we should expect on average for a Poisson deployment of fixed rate. However, any specific deployment may deviate from the average, i.e., there is a non-zero (although very small) probability that some small number of vectors are linearly dependent. The lemma effectively enables use of eq. (2.12) and (2.13) for computation of orthogonal projection errors, whenever the underlying deployment pattern is Poisson.

2.6 Related Work

The ultimate aim of sampling-interpolation applications is to obtain spatiotemporal models of the sensed physical phenomena. The unique characteristic of WSNs in this respect is that such models should be built *efficiently*, i.e., taking into account the limited resources of the network. Some related efforts do exist in the sensor networks' literature [Gue04], [Per04], [Zha03]. However, they generally assume some specific kind of prior knowledge on the underlying physical phenomenon. The hexagon based sensor reporting scheme [Zha03] requires exact knowledge of the second-order statistics of the spatial process beforehand, while the blue noise sampling technique [Per04] needs similar information to construct a blue noise spatial filter. The distributed regression framework [Gue04] performs in-network spatial data modeling by computing weights of local basis functions after

sensor measurements have been partitioned with kernels. The number and size of the kernels as well as basis functions used by each, in essence model correlation among various positions on the field and assume therefore a rough, prior knowledge of how this correlation varies over it. Our aim in the present work is to make as few a-priori assumptions about the statistics of the physical phenomenon as possible.

However, transcending the specific sensor networks regime, in the last few decades, modeling of space-time data has actually been the target of tremendous amounts of research effort in the environmental sciences, i.e., the atmospheric, oceanographic and geological sciences. Excellent overviews of related techniques can be found in [Bow97], [Har04], [Has90], [Le06], [Pac03]. These can be roughly divided to *regression models* and *random process* models.

Regression models generally assume that a data and location dependent deterministic structure underlies the observations and regard random effects as white noise. They include such subject areas as kernel regression, spline smoothing and orthogonal series expansions. Wavelet expansions and associated multi-resolution analyses have especially attracted a lot of interest during recent years. Although such models have a richly developed theory behind them, they can lead to interpretation problems when used in a sensor networks context. For example, the notion of correlation between two or more sensors is not well defined in such a framework. Our intuition is that sensor networks can benefit more from models in which the effects of individual sensors can be decoupled, if necessary, within the model and where the physical meaning of spatial locations and measurements is well preserved. As a remark, it is noted that all of these methods are instantiations of *generalized linear*

models which are currently at the forefront of machine learning research. Another development in this area is compressive sensing [Baj06], [Dua06], [Wan07], which will be discussed in the next chapter.

Random process models on the other hand, assume that observations belong to an underlying space of random functions. The functions are sampled at the monitoring sites. Exactly because these methods explicitly view measurements gathered at individual sites as random variables, they seem to be more naturally suited for interpretation of sensor network data. Relevant examples include, but are not limited to, single time instant geostatistical kriging, empirical orthogonal functions, and hierarchical Bayesian modeling. Hierarchical Bayesian modeling in particular, is the subject of ongoing research. It relies on a view of the physical phenomenon as a Gaussian process in space and time. The term 'Gaussian Process' (GPs) is used in the literature to describe a random process, for which the joint probability distribution considered at any finite number of index points (i.e., locations and time instants) is Gaussian. Generally, trends and covariance functions form the first level of the hierarchy while their uncertainty, i.e., constituent parameters and prior probability distributions on these parameters, forms the second level of the hierarchy. The goal is to obtain posterior distributions on quantities of interest, after *data have been observed*. Sampling of posterior distributions, often referred to as *Bayesian inference*, can be well served by Markov Chain Monte Carlo techniques.

However, for our particular objective of spatial interpolation with a WSN, even these methods are not without shortcomings. The main drawback is that the statistical correlation structure assumed is usually inseparable in space and time, i.e., spatial

effects cannot often be decoupled from temporal ones. As a result, spatial structure may be lost or altered when subtracting temporal trends and vice versa. By contrast, we make the more conservative assumption of a mean constant in time at each spatial location. In addition, these methods are burdened with considerable computational complexity. In large scale meteorology for example, they aim at explaining data accumulated at a small number of sites and over several years. Their applicability however, in scenarios with real time requirements, incorporating thousands of sensors and lacking prior measurements is not obvious. The methods that will be presented in subsequent chapters are simpler and faster in comparison.

A subclass of hierarchical Bayesian models which is perhaps closer to what we have presented in this chapter and has been used in a sensor networks' context is Gaussian processes in space [Gue05], [Ras06]. Since random variables are involved, Gaussian processes in general and spatial ones in particular lie in a Hilbert space, exactly as has been discussed in this chapter. The major difference is that the framework presented here does not rely on the fact that the underlying distributions are Gaussian and therefore encompasses Gaussian processes as a special case. Our methods are potentially applicable to any type of distribution, as long as estimators of the type (2.1) can be considered relevant.

2.7 Summary

In this chapter, we have outlined a view of spatial interpolation based on the notion of a suitably chosen Hilbert space. Within the framework of this Hilbert space, each sensor can be represented by a vector characterized by the sensor's spatial

position. Correlations among sensors are captured by the geometry of the space, i.e., its inner product. Moreover, the quality of interpolation can be directly linked to orthogonal projection error in this space. Finally, when the deployment pattern can be regarded as random, in the sense that no particular region of the field is given preference over another for deployment purposes, explicit expressions can be invoked to compute such errors.

2.8 Acknowledgements

This chapter is, in part, a reprint of material published in the article: “Leveraging Redundancy in Sampling-Interpolation Applications for Sensor Networks”, appearing in the Proceedings of the Third IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2007, Lecture Notes in Computer Science 4549, Springer 2007, ISBN 978-3-540-73089-7. It is also, in part, a reprint of material submitted for publication in the Springer Journal of Signal Processing Systems, under the title “Efficient Sensing Topology Management for Spatial Monitoring with Sensor Networks”. In both cases, I was the primary researcher and author and Curt Schurgers supervised the research.

CHAPTER 3

3. SPATIAL INTERPOLATION: THE REACTIVE CASE

3.1 Background

In this chapter, we are going to examine solutions to the reactive management scenario of Section 1.6. Consider the example of a wireless sensor network (WSN) monitoring radiation level in a contaminated area. In such a scenario, inexpensive sensors are likely to be dropped from an airplane over the field of interest in an ad hoc manner. In addition, we, the network practitioners, do not have a concrete way of knowing beforehand what to expect in terms of variation of the monitored values over the whole area. This lack of prior knowledge also hinders exact characterization of the sensing behavior of devices before deployment.

However, a practical strategy that can be utilized is to initially overdeploy: make a pessimistic assumption on the necessary number of sensors and throw such number out. The problem essentially becomes that of *managing which sensors are needed, out of all deployed, so as to achieve an accurate enough interpolation over time*. After initial deployment, some sensors will be redundant from an application point of view. In a sampling-interpolation setting this essentially translates to oversampling: only a few of the gathered samples are needed to provide a sufficiently good interpolation. The challenge is then how to take advantage of this redundancy and achieve longevity of operation for the network as a whole. The network practitioner essentially trades the

cost of overdeployment with improving the network in another dimension, namely that of lifetime.

A straightforward way to leverage redundancy during operation is by keeping only a subset of sensors sensing and communicating samples. The subset is such that samples collected from it can still provide a sufficiently good interpolation. The key idea towards overall lifetime gains, however, is devising not just one, but multiple subsets of sensors, which are, ideally, disjoint and each individually achieves the desired interpolation fidelity [Kou06], [Lia06], [Sli06]. At every point in time only one such subset is made active and produces sensing data packets. Sensors in all other subsets may be in an energy efficient sleep state or forwarding such data, albeit not generating data themselves. The data producing functionality is then rotated amongst subsets.

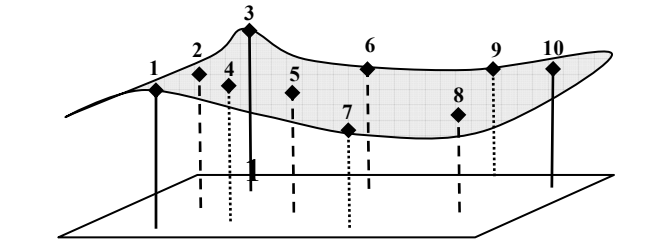

Figure 3.1: Rotating subsets of active sensors

The effectiveness of this *sensing topology management* scheme for obtaining lifetime gains, assuming subsets are disjoint, depends upon the number of subsets: the more subsets available the higher the proportional increase in lifetime. Figure 3.1 shows interpolation being performed sequentially with three different subsets of sensors. If these are chosen so that interpolation with any one of them alone is

accurate enough, as the phenomenon changes over time, the sensor network can be effectively thought of as being three superimposed networks, all basically equivalent with respect to interpolation. Ideally, this may lead to a three-fold increase in network lifetime.

3.2 Optimization Goals

The purpose of this section is to elaborate on the notion of "accuracy" for spatial interpolation. The setup assumed is exactly the same as in section 2.2: a static network of N sensors scattered over an observation field F.

Interpolation is inherently approximate since the network consists of a finite number of sensors and, additionally, the measurements are themselves affected by various factors such as measurement noise, quantization noise etc. In order to characterize the quality of interpolation, a distortion measure has to be introduced. Here, we explore two such measures:

$$\underset{\boldsymbol{x}\in F}{avg}(D_{\boldsymbol{m}_k}) = \frac{1}{|F|}\int_{\boldsymbol{F}} E[(\hat{S}_k(\boldsymbol{x},t_i) - S(\boldsymbol{x},t_i))^2]d\boldsymbol{x} \le D \tag{3.1}$$

$$\max_{\boldsymbol{x}\in F}(D_{\boldsymbol{m}_k}) = \max_{\boldsymbol{x}\in F}(E[(\hat{S}_k(\boldsymbol{x},t_i) - S(\boldsymbol{x},t_i))^2]) \le D \tag{3.2}$$

where D is a user defined distortion criterion. Inequality (3.1) defines distortion as spatially averaged Mean Squared Error (MSE), while inequality (3.2) defines distortion as the maximum MSE over space. There is a fundamental difference between these two criteria: the latter demands that the MSE of interpolation be less than the target D at *all* locations in the observation field while the former can tolerate large MSE at some locations if it is compensated by low MSE at other locations.

The choice of optimization criterion belongs to the steering user application. Spatially averaged distortion is plausible in many cases and has been used extensively utilized in the domain of geostatistical studies [Dig06], [Zhu06]. However, it does not always capture local features of the physical phenomenon, which are of interest in many practical scenarios. For example, a situation where distortion is high in a region with few sensors and very low in a region with many sensors would be overlooked by the spatially averaged distortion criterion. Qualitatively, these two different definitions of distortion will be tackled with different sensing topology management algorithms, as will be shown in the following sections.

A subset of sensors is acceptable for a particular application, if the distortion associated with it by virtue of the chosen criterion (i.e. (3.1) or (3.2)) does not exceed the threshold D. Since the network consists of N sensors, it may be the case that more than one disjoint subsets of sensors exist, all of which adher to this distortion bound. *The overall goal of our sensing topology management scheme is precisely to partition sensors in as many disjoint subsets as possible, while still meeting a desired distortion bound for each subset*. A *monitoring schedule* is then a sequence of sensor selections, i.e., subsets, over time.

3.3 Sensing Topology Management - Fundamentals

Presenting a novel practical way to devise disjoint subsets for both distortion criteria (3.1) and (3.2) is the primary goal of this chapter. Our novel contribution lies in how to select the sensors comprising each subset. The proposed selection algorithms are to be executed centrally. Since data interpolation, is performed at the

application end-point (data fusion center) anyway, it is not unreasonable to have sensor selection done at this center as well. The important goal is to effectively reduce the number of data packets which have to be communicated through the network. Although the exact energy savings achieved by our schemes will depend on the actual MAC protocol and routing used, reducing the number of data producing nodes is beneficial in exactly the same manner as it is for the case of event detection applications [Abr04], [Sli01]. Furthermore, the only energy overhead of our schemes compared to an unscheduled network comes from switching between monitoring subsets. This energy cost can be kept to a minimum for example with randomized flooding [Kin06] of a bit-mask packet denoting subset membership, whenever such a switch is necessary. A more meticulous account of potential energy savings for the schemes presented here will be given in Section 3.9.

Before we outline our specific contributions in the next sections, we briefly describe related practicalities. Specifically, notice that inequalities (3.1) and (3.2) are based on a statistical characterization of the underlying random process which is not available a priori. Instead it has to be learned after network deployment. A two-phase strategy addresses this problem:

1. During the *learning phase*, all N sensors report their data, where, in addition to interpolation itself, the goal is to estimate relevant statistical properties of the process. This learning phase can extend over multiple time instants $t_{i...\Theta+i}$, during which the monitoring application is fully operational, albeit not in a manner that takes advantage of redundancy. In addition, it is possible that statistics of the process change over the course of time (for example the

temperature readings of two sensors may show different correlation properties in summer as compared to winter), dictating that the learning phase may have to be repeated periodically.

2. During the *monitoring phase*, only sensors from an active subset report and subsets are rotated over time.

To devise monitoring subsets that optimize criteria (3.1) and (3.2), it is essential that for any given subset of sensors $\{S(\boldsymbol{X}_k^p, t_i)\}$, p=1…$|\boldsymbol{m}_k|$, the values of the MSEs contained in those expressions can be computed. Recall from section 2.4. that the subset of sensors $\{S(\boldsymbol{X}_k^p, t_i)\}$, p=1…$|\boldsymbol{m}_k|$, defines a finite dimensional subspace H_{Xk}. For the *best linear interpolation*, such MSEs are given by eq. (2.12) or (2.13). However, actual computation of the error of optimal interpolation, i.e., the error of orthogonal projection onto the related subspace, requires knowing the inner product of $S(\boldsymbol{x}, t_i)$, $\forall \boldsymbol{x} \in F$, with all the sensor vectors. From eq. (2.8), this is equivalent to knowing the entire continuous spatial covariance function:

$$< S(\boldsymbol{x},t_i), S(\boldsymbol{X}_k^p,t_i) >= C(\boldsymbol{x},\boldsymbol{X}_k^p) \tag{3.3}$$

Although it is possible to estimate the continuous covariance function [Kra06], [Not02], this is ultimately a costly and intricate procedure that also does not lend itself to providing distortion guarantees. Instead, we restrict $S(\boldsymbol{x}, t_i)$ to elements of a subspace of H_S, namely that spanned by all deployed sensors H_{X0}. We refer to H_{X0} as the *primary subspace*. It is generated by $\{S(\boldsymbol{X}_0^p, t_i)\}$, $p=1..N$, and is therefore of dimension at most N. Essentially, we assume that the initial number of sensors is large enough so that H_{X0} is a close approximation to H_S, or in other words that the sensors

can capture enough details of the underlying physical phenomenon in the first place. The distortion metric is thus defined in relation to the maximum information we could extract with our initial deployment. Formally, it means that eq. (3.1) and (3.2) are transformed into:

$$\underset{\boldsymbol{x}\in F}{avg}(D_{\boldsymbol{m}_k}) \approx \frac{1}{|F|}\int_{\boldsymbol{F}} E[(\hat{S}_k(\boldsymbol{x},t)-\hat{S}_0(\boldsymbol{x},t))^2]d\boldsymbol{x} \le D \tag{3.4}$$

$$\max_{\boldsymbol{x}\in F}(D_{\boldsymbol{m}_k}) \approx \max_{\boldsymbol{x}\in F}(E[(\hat{S}_k(\boldsymbol{x},t_i)-\hat{S}_0(\boldsymbol{x},t_i))^2]) \le D \tag{3.5}$$

A first practical point is that during the learning phase any sensor selection algorithm needs to actually evaluate eq. (3.4) or (3.5) in order to assess its performance. Since for a real system the expectation operator cannot be known, it is approximated with an average over W time instants, where $W \le \Theta$. During the learning phase, all N sensors report their data for Θ time instants. Thus each sensor collects a time series of Θ values $\{S(\boldsymbol{X}_0^p, t_i)\}_{i=1..\Theta}$. By virtue of mean square ergodicity, surfaces $\{\hat{S}_0(\boldsymbol{x}, t_q)\}_{q=1...\Theta}$, obtained by interpolating these values for fixed time instants can be used as the best available approximation to the ground truth (i.e., they can thought of as *reference surfaces*):

$$\underset{\boldsymbol{x}\in F}{\hat{avg}}(D_{\boldsymbol{m}_k}) = \frac{1}{|F|}\int_{\boldsymbol{F}} \frac{1}{W}\sum_{i=1}^{W}(\hat{S}_k(\boldsymbol{x},t_i)-\hat{S}_0(\boldsymbol{x},t_i))^2 d\boldsymbol{x} \le D \tag{3.6}$$

$$\underset{\boldsymbol{x}\in F}{\hat{\max}}(D_{\boldsymbol{m}_k}) \approx \max_{\boldsymbol{x}\in F}(\frac{1}{W}\sum_{i=1}^{W}(\hat{S}_k(\boldsymbol{x},t_i)-\hat{S}_0(\boldsymbol{x},t_i))^2) \le D \tag{3.7}$$

A second practical point is that to evaluate orthogonal projection error, eq. (2.12) and (2.13) require knowledge of the inner products between all sensors. From eq. (2.2) these inner products correspond to covariances and can be readily estimated after the learning phase as follows. Let B_0 be the ΘxN matrix formed by stacking the time

series produced by individual sensors together as columns:

$$B_0 = \begin{bmatrix} S(\boldsymbol{X}_0^1, t_1) & \dots & S(\boldsymbol{X}_0^N, t_1) \\ \dots & \dots & \dots \\ S(\boldsymbol{X}_0^1, t_\Theta) & \dots & S(\boldsymbol{X}_0^N, t_\Theta) \end{bmatrix} \tag{3.8}$$

We will refer to B_0 as the *data matrix*. Firstly, a vector estimate of the spatial mean function $\hat{\boldsymbol{\mu}}_0 = \{\mu_0^p\}_{p=1\dots N}$ evaluated at the sensor locations is obtained from B_0. This can be done for example by averaging the columns of B_0, since the spatial mean function is assumed to be constant in time (see eq. (2.5)). Then the empirical covariance matrix provides an estimate of the inner products:

$$\hat{G}_0 = \frac{1}{\Theta - 1} \cdot (B_0 - \hat{\mu}_0)^T \cdot (B_0 - \hat{\mu}_0) \tag{3.9}$$

with $\hat{\mu}_0$ being a matrix with all rows equal to $\hat{\boldsymbol{\mu}}_0$. The matrix $\hat{G}_0$ will also be referred to as the approximate Grammian matrix of inner products among all deployed sensors. Equation (3.9) converges to the true covariances for Θ large enough, again by virtue of mean square ergodicity. A subtle point is that the estimate of eq. (3.9) results in a Grammian of full rank (i.e. of rank N) only if $\Theta \geq N$, i.e., only if the number of measurements in time is at least as large as the number of sensors. It should be stressed however that eq. (3.9) represents only one possible way of estimating the covariances from the data matrix, used in this dissertation for its simplicity. Other more sophisticated and robust methods of estimation exist in the literature for this purpose [Sch05] and are equally applicable. In Section 3.9 we will briefly examine the effect of a small number of measurements on the performance of our schemes.

3.4 Complexity of Optimal Solution

In the Hilbert space framework, introduced in the previous section, each individual sensor can be thought of as a vector in the primary subspace. Consequently, the sensing topology management problem of selecting disjoint subsets of sensors translates into selecting subsets of vectors instead. The goal is to maximize the number of subsets that can be found (or equivalently, minimize the average number of vectors in each subset), while ensuring that each subset can provide a sufficiently accurate interpolation, in terms of the chosen distortion definition.

As a first step in tackling this problem, we consider a more basic variant, namely that of finding just a single minimal subset: *Given an initial set of sensors, find the minimal subset which yields an interpolation that has an expected distortion of at most D.* Finding multiple subsets, is a generalization of this problem and hence computationally at least as hard. Furthermore, our algorithm will be built on a good understanding of solutions for the single subset selection problem.

The basic single-subset problem can essentially be seen as finding an approximate basis for the primary subspace. The term 'basis' for a V-dimensional Hilbert, will hereafter refer to any set of V linearly independent vectors that span the space. Note that the vectors forming a basis need not be orthogonal to each other. As such, the problem is related to the problem of sparse signal approximation with general dictionaries, which has been studied in signal processing literature [Cot05], [Dav97], [Don03], [Rao03], [Tro06], [Wip04], [Wip07]. The term 'dictionary' will hereafter refer to a set of non-orthogonal vectors used for representation in a Hilbert

space, without necessarily forming a basis for that space. The sparse approximation problem in a V-dimensional Hilbert space is how to select the best vectors out of a redundant dictionary of size $P \geq V$ to approximate a given target vector (simple sparse approximation) or multiple target vectors (simultaneous sparse approximation) in the space. This ideally requires enumerating all possible subsets of vectors, an operation of which the cost is exponential in P [Dav97], [Don03]. In the case of a general dictionary, the resulting computation is provably NP-hard [Dav97].

In our scenario, the dimension of the primary subspace H_{X0} is at most N. The set of N vectors that correspond to the initially deployed sensors $\{S(\boldsymbol{X}_0^p, t_i)\}_{p=1..N}$ is therefore a redundant dictionary for this space. Given a target vector $S(\boldsymbol{x}, t_i)$, a dictionary of size N effectively means that the computational cost to optimally select sensors to approximate it grows exponentially with the size of the network N. For our particular case we have also proved a stronger result: that for Poisson deployments, the vectors $\{S(\boldsymbol{X}_0^p, t_i)\}_{p=1...N}$ are linearly independent on the average. Linear independence means that the dimension of H_{X0} is N, i.e., the dictionary $\{S(\boldsymbol{X}_0^p, t_i)\}_{p=1...N}$ is also a basis for the space, rendering optimization over *any* redundant dictionary for this space exponentially hard with the size of the network. For more details refer to section 2.5.

Since the single-subset problem is hard, the same will hold for the extended problem of finding multiple subsets. As a result, we have to resort to heuristic approaches to perform the selection of multiple active subsets of sensors.

3.5 Jittered Grid Sampling

3.5.1 Rationale

We will first present subset selection schemes pertaining to the spatially averaged distortion criterion (3.6). Optimizing the spatially maximum distortion criterion (3.7) is a more intricate problem and will be treated in a separate subsection (specifically, Section 3.8).

A first conceptual foundation to efficiently partition sensors into active subsets is the observation that spatial variation for many physical processes of interest (e.g. temperature, humidity, salinity, acidity) can in practice be well approximated by models that are structurally regular throughout the observation field. Researchers in environmental sciences (e.g. atmospheric sciences, geostatistical monitoring) have in fact consistently modeled such physical quantities as spatially stationary processes [Dig06], [Ste99]. Structural regularity in space has been expressed through a multitude of ways in the sensor networks regime as well: sparse representation in N-dimensional bases [Baj06], Gaussian processes [Kra06] or bandlimited processes [Bal07], [Per04] are the most commonly adopted ones. With a spatially stationary model for the physical phenomenon of interest, regular grid designs, specifically rectangular and triangular grids, are frequently used to determine the monitoring sites [Dig06], [Zhu06]. In fact, an abundance of theoretical results exists for the distortion performance of such designs when the underlying statistics are assumed Gaussian [Ste99].

In our setup, sensors are chosen from a pre-deployed set which does not necessarily form a regular grid. The key idea is to try to approximate a grid structure

instead. To do this we impose a *virtual grid* over the observation field and then 'map' a subset of sensors onto this grid. The mapping should satisfy some closeness criterion. By offsetting the virtual grid the mapping procedure can be repeated a number of times so as to obtain many different sensor subsets. As an example, Figure 3.2 shows a virtual square grid of 16 points superimposed on a network of 50 sensors.

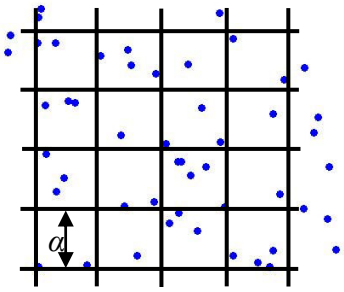

Figure 3.2: A virtual grid imposed on a random network

For simplicity, but without loss of generality, we will hereafter focus on square grid structures. It should be stressed that, ultimately, the method for subset construction described here can be used to obtain sensor subsets based on any type of chosen grid structure, i.e., not necessarily a square grid. Other choices could be triangular or hexagonal grids, which are known to be nearly optimal for certain forms of stationary correlations [Kun03], [Zhu06]. In our case where the actual statistics of the physical process are unknown, i.e., the optimal sampling structure is unknown, square grids represent a pragmatic choice that gives good interpolation results. We will demonstrate this in Section 3.9.

3.5.2 Subset Construction

A virtual square grid is uniquely defined by its size, or equivalently its side α, as shown in Figure 3.2, while grids of the same size are distinguished by their offsets relative to a fixed point (x_0, y_0). For results presented in this dissertation, we offset grids along their common diagonal. Supposing that a series of virtual square grids are available, a naive strategy to construct sensor subsets would be the following:

Pure Jittered Grid Sampling (PJiG): Construct monitoring subsets by sequentially selecting the sensors that are closest in Euclidean distance to grid points.

The *PJiG* scheme is a useful tool for theoretical analyses; however, in practical settings it has one important drawback: it does not necessarily produce disjoint subsets of sensors. Consider a virtual grid that consists of M points. Then, according to the pure jittered equidistant sampling rule, a single sensor may be the closest one to more than one grid points. At most (but maybe not exactly) M sensors will be selected based on such a grid. This characteristic of *overlap* or *collision* is possible within the same subset or across subsets and is undesirable for our objective of increasing lifetime.

The crucial element for subset construction is then how to obtain *disjoint* subsets from virtual grids. Essentially a 1-1 mapping is needed, such that each sensor is mapped onto a grid point only once and optimality is achieved with respect to the closeness criterion. To tackle this, suppose that a total of $\lfloor N/M \rfloor$ virtual grids have been constructed, consisting of $\lfloor N/M \rfloor \cdot M$ points. We construct a complete bipartite graph (U, V, E) consisting of a set of vertices U corresponding to the grid points, a set

of vertices V corresponding to the sensors and a set of edges $E = \{[u_i, v_j]\}$ with weights w_{ij} equal to the Euclidean distance between grid point i and sensor j. If the number of grid points $\lfloor N/M \rfloor \cdot M$ is less than the number of sensors N (i.e., if N is not an integer multiple of M), the set U of grid points is enhanced by adding to it $N - \lfloor N/M \rfloor \cdot M$ points with incident edges of infinite weight. With this formulation, the problem of mapping sensors onto grid points is equivalent to the minimum weight perfect matching problem for bipartite graphs and can be efficiently solved by the Hungarian Algorithm [Pap98]. We call the overall procedure *Jittered Grid Sampling* (*JiG*):

Jittered Grid Sampling (JiG): Construct subsets indexed $k = 1 \ldots \left\lfloor \frac{N}{M} \right\rfloor$, with application of the Hungarian algorithm on the matrix $\{w_{ij}\}$ of weights defined by the Euclidean distances separating grid node i, $i=1 \ldots N$ from sensor j, $j=1 \ldots N$.

As a remark, note that for spatially stationary processes, Euclidean distance directly translates to spatial correlation. In this sense, edges in the bipartite graph (U, V, E) are weighed by the magnitude of correlation between grid points and sensors as points in 2-D space.

An important technical issue that remains is how to select the size M of the virtual square grids. This can be accomplished with reference surfaces $\{\hat{S}_0(\mathbf{x}, t_q)\}_{q=1 \ldots \Theta}$ constructed during the learning phase (see section 3.4). For a single dimensional network, we have proposed to estimate the spatial Nyquist rate of the physical

phenomenon from these reference surfaces [Lia06], which essentially boils down to spectrum estimation. For two dimensional networks examined here, 2-D Nyquist rate estimation from reference surfaces is also possible [Lim90] and a multitude of related spectrum estimation methods exist [Sto05], [Tho01], [Wae99]. However, we have found that empirical determination of associated parameters is not as straightforward as in the single dimensional case and depends significantly on the underlying physical process and estimation method used. Another drawback is that the Nyquist rate on a finite field is not necessarily optimal in terms of sampling and interpolation distortion [Kun03].

Instead, we have used eq. (3.6) as a performance criterion. To choose the appropriate grid size $M = n^2$, the integer n *is* iteratively increased until the resulting sensor subsets satisfy the distortion bound as computed through eq. (3.6). Furthermore, virtual grids are offset along their common diagonal. The complete sensing topology management scheme based on *JiG* is shown in Figure 3.3.

1 Using data from the learning phase, construct W reference surfaces $\{\hat{S}_0(\mathbf{x}, t_q)\}_{q=1\ldots W}$.
2 Select n, set $M = n^2$.
3 Construct sets indexed $k = 1\ldots\lfloor N/M \rfloor$, with application of the Hungarian algorithm on the matrix $\{w_{ij}\}$ of weights defined by the Euclidean distances separating grid node i, $i=1\ldots N$ from sensor j, $j=1\ldots N$.
4 Compute $\underset{\mathbf{x}\in F}{a\hat{v}g}(D_{\mathbf{m}_k})$, from eq. (3.6) for all of these sets. If $\underset{\mathbf{x}\in F}{a\hat{v}g}(D_{\mathbf{m}_k}) > D$ for at least one of them, go to step 2 with $n = n + 1$, otherwise finalize the M computed in step 2.
5 The $N - \left\lfloor \frac{N}{M} \right\rfloor \cdot M$ sensors corresponding to dummy grid points are randomly distributed among devised sets.

Figure 3.3: Jittered Grid (*JiG*) subset construction algorithm

3.5.3 Performance Analysis

In the previous subsection we described a method for devising monitoring subsets of sensors based on a jittered grid sampling notion. The key idea is emulation of sampling on a regular grid. If the physical process statistics are considered fixed, the distortion performance of jittered grids will depend on the characteristics of the jitter and the interpolation scheme used. This subsection provides an analysis of the distortion performance of this sensing topology management scheme, drawing on the Hilbert space framework for interpolation presented in Chapter 2.

3.5.3.1 Assumptions

For our analysis, we make the following assumptions:

- Sensors are assumed to be scattered in a Poisson fashion over the observation field. For a large fixed number of sensors N, we approximate the rate of the Poisson process with the density of the network:

$$\beta = \frac{N}{|F|} \tag{3.10}$$

- When the sensor closest to a grid point is chosen, the point process describing the scattered sensors is purely Poisson and in particular, Poisson of constant rate β. This is the case when each sensor selection is performed independently of all previous such selections, i.e. the selection process is *memoryless*. Thus our analysis pertains to *PJiG*. *JiG* on the other hand, is a scheme with memory: selections are engineered to be non-colliding by finding a perfect matching. Because of this strict memory effect the resulting point process is no longer

Poisson, rendering its analytical study intractable. However, the central theme of our approach is well captured by the memoryless variant for dense networks.

- The monitored random process is spatially stationary. This implies that the spatial covariance of eq. (2.8) is a function solely of the difference of spatial indices:

$$< S(\boldsymbol{x}_1, t_1), S(\boldsymbol{x}_2, t_2) >= E[(S(\boldsymbol{x}_1, t_1) - \mu(\boldsymbol{x}_1)) \cdot (S(\boldsymbol{x}_2, t_2) - \mu(\boldsymbol{x}_2))] = C(\boldsymbol{x}_1 - \boldsymbol{x}_2) \tag{3.11}$$

3.5.3.2 Jitter Formulation

A first key step in what follows is deriving the distribution of the jitter, i.e. how far away the sensors selected are from points of the grid. Denoting a node of the grid with *O*, let *A* denote the sensor closest to it, as depicted in Figure 3.4.

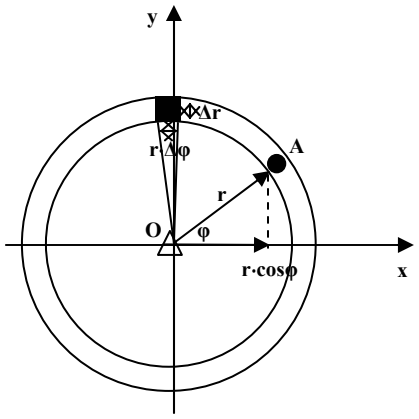

Figure 3.4: Jitter formulation

Under the above assumptions, the distribution of the magnitude of the distance to the closest sensor, *r*, can be derived based on the notion of 'hitting' or 'vacancy' probabilities in stochastic geometry:

$$\Pr[r \le z] = \Pr[(circle(O,z)\ is\ empty\ of\ \ sensors)^{C}] =$$

$$= 1 - \Pr[circle(O,z)\ is\ empty] = 1 - \exp(-\beta \cdot \pi \cdot z^2) \tag{3.12}$$

where $(\cdot)^C$ denotes the complement of an event. To fully characterize the position of *A*, i.e. the jitter, the joint distribution of r and φ must be found. Consider the joint probability of the event $((r_0 \le r \le r_0 + \Delta r) \cap (\varphi_0 \le \varphi \le \varphi_0 + \Delta\varphi))$:

$$\Pr[(r_0 \le r \le r_0 + \Delta r) \cap (\varphi_0 \le \varphi \le \varphi_0 + \Delta\varphi)] =$$

$$\Pr[\varphi_0 \le \varphi \le \varphi_0 + \Delta\varphi \mid r_0 \le r \le r_0 + \Delta r] \cdot \Pr[r_0 \le r \le r_0 + \Delta r] = \frac{\Delta\varphi}{2 \cdot \pi} \cdot \Pr[r_0 \le r \le r_0 + \Delta r] \underset{\Delta\varphi \to 0}{\overset{\Delta r \to 0}{\Rightarrow}}$$

$$f_{r\varphi}(r,\varphi) = \frac{1}{2 \cdot \pi} \cdot 2 \cdot \pi \cdot \beta \cdot r \cdot \exp(-\beta \cdot \pi \cdot r^2) = f_{\varphi}(\varphi) \cdot f_r(r) \tag{3.13}$$

where $f_{(\cdot)}(\cdot)$ denotes the probability density function of the involved quantity. For the analysis of distortion in the next subsection the distribution of the Cartesian coordinates of the jitter as shown in Figure 3.4 is also needed. It can be derived as follows:

$$\Pr[r \cdot \cos\varphi \le z] = \iint\limits_{r \cdot \cos\varphi \le z} \beta \cdot r \cdot e^{-\beta \cdot \pi \cdot r^2}\, dr \cdot d\varphi \underset{y = r \cdot \sin\varphi}{\overset{x = r \cdot \cos\varphi}{=}} \int\limits_{-\infty}^{z} \sqrt{\beta} \cdot e^{-\beta \cdot \pi \cdot x^2}\, dx \int\limits_{-\infty}^{\infty} \sqrt{\beta} \cdot e^{-\beta \cdot \pi \cdot y^2}\, dy$$

$$= \int\limits_{-\infty}^{z} \frac{1}{\sqrt{2 \cdot \pi}} \cdot \sqrt{2 \cdot \pi \cdot \beta} \cdot e^{-\frac{1}{2}(x \cdot \sqrt{2 \cdot \pi \cdot \beta})^2}\, dx \tag{3.14}$$

The last part of eq. (3.13) says that the Cartesian coordinate $x = r \cdot \cos\varphi$ has a Gaussian distribution, specifically $N(0, \frac{1}{2 \cdot \pi \cdot \beta})$, where $N(\mu, \sigma^2)$ denotes the Gaussian distribution

with mean μ and variance σ^2. Computation of the y-coordinate leads to an exactly symmetric integration, so that both Cartesian coordinates have the same Gaussian distribution. Based on the fact that the polar coordinates are independent and using standard probability tools [Sta02] it can be proved that the Cartesian coordinates are independent as well.

3.5.3.3 Expected distortion

In order to analyze expected distortion we further need to analyze interpolation, i.e. the exact procedure by which a spatial profile is obtained based on a subset of sensors. This can be done based on the Hilbert space framework of Chapter 2. Equation (2.13) effectively describes expected distortion at any fixed position $\boldsymbol{x}$ on the field in terms of the spatial covariance function $C(\cdot)$ of the underlying random process. Our ultimate purpose is to statistically characterize the orthogonal projection error for all Poisson deployments of a certain rate β, with the underlying random process, i.e., $C(\cdot)$, regarded as fixed. Essentially, what follows enables investigation of expected distortion with respect to network density for any fixed physical process model. For notational convenience, in what follows we will drop the indices k referring to subset $\boldsymbol{m}_k$ as well as call the orthogonal projection with a subset of sensors, Γ and the number sensors in the subset, M. Equation (2.13) then becomes:

$$\Gamma = C(0) - \frac{(\sum_{p=1}^{M} \left| C(\boldsymbol{\eta} - \boldsymbol{X}^p) \right|^2)^2}{\sum_{p=1}^{M}\sum_{q=1}^{M} C(\boldsymbol{\eta} - \boldsymbol{X}^p) \cdot C(\boldsymbol{\eta} - \boldsymbol{X}^q) \cdot C(\boldsymbol{X}^p - \boldsymbol{X}^q)} \tag{3.15}$$

Notice that Γ is a random variable dependent on the deployment of sensors: for any given spatial covariance structure $C(\cdot)$, for any particular deployment $\{(x_1^p, x_2^p)\}$, $p = 1 \ldots M$ and for any target coordinates (η_1, η_2), the quantities $\boldsymbol{\eta} - \boldsymbol{X}^p$ and thus eq. (3.15) can be directly computed. With fixed coordinates (η_1, η_2), Γ can in fact be viewed as a non-linear function $h(\cdot)$ of the following collection of $2 \cdot M$ constituent random variables:

$$\begin{aligned} G_1 &= \eta_1 - x_1^1 \\ &\vdots \\ G_M &= \eta_1 - x_1^M \\ G_{M+1} &= \eta_2 - x_2^1 \\ &\vdots \\ G_{2 \cdot M} &= \eta_2 - x_2^M \end{aligned}$$

$$\Gamma = C(0) - \frac{(\sum_{k=1}^{M} C^2(G_k, G_{M+k}))^2}{\sum_{k=1}^{M}\sum_{n=1}^{M} C(G_k, G_{M+k}) \cdot C(G_n, G_{M+n}) \cdot C(G_k - G_n, G_{M+k} - G_{M+n})} = h(G_1, \ldots, G_{2 \cdot M}) \tag{3.16}$$

where we have used the convention that $R(\boldsymbol{\eta} - \boldsymbol{X}^p) = R(\eta_1 - x_1^P, \eta_2 - x_2^p)$. By virtue of eq. (3.14) for *PJiG*, each of the G_k follows a Gaussian distribution. The mean is the difference between the Cartesian coordinates of η i.e. (η_1, η_2) and those of the grid point (u_1^p, u_2^p) corresponding to sensor $\boldsymbol{X}^k$. The variance is $\frac{1}{2 \cdot \pi \cdot \beta}$. In addition, all G_k are independent of each other: G_k is independent of G_n for $k, n < M$ because they correspond to hitting probabilities of a Poisson process centered at different points on the field and G_k is independent of G_{M+k} because they correspond to the Cartesian

coordinates of the jitter, which have been proved to be independent in Section 3.5.3.2. Hence the covariance matrix of these variables will be diagonal:

$$\begin{gathered}\boldsymbol{G} = (G_1,...G_{2\cdot M}) \propto N(\boldsymbol{\mu}, \Sigma) \\ \boldsymbol{\mu} = (\eta_1 - u_1^1,...,\eta_1 - u_1^M,\eta_2 - u_2^1,...,\eta_2 - u_2^M) \\ \Sigma = \frac{1}{2\cdot\pi\cdot\beta}\cdot\boldsymbol{I}\end{gathered} \tag{3.17}$$

It is useful to point out that the variance of these random variables is inversely proportional to the network density β, i.e. the denser the network the better the approximation of a virtual grid obtained.

Through eq. (3.16) and (3.17), Γ can be viewed as a non-linear transformation of a multi-dimensional Gaussian distribution. By studying the statistical behavior of Γ for different covariance structures of the underlying physical phenomenon, i.e., for different functions $C(\cdot)$, we can gain useful insights on how distortion performance scales with network density. For covariance models of interest [Ber01], the exact distribution of Γ is hard to derive analytically. However, numerical methods can be used to approximate the mean and variance of Γ, which are what interests us ultimately. The first two moments of Γ can be expressed as:

$$\begin{aligned}E[\Gamma] &= \int_{-\infty}^{\infty} h(\boldsymbol{G})\cdot P(\boldsymbol{G})d\boldsymbol{G} \\ E[\Gamma^2] &= \int_{-\infty}^{\infty} h^2(\boldsymbol{G})\cdot P(\boldsymbol{G})d\boldsymbol{G}\end{aligned} \tag{3.18}$$

where $P(\boldsymbol{G})$ is the probability density function of the multi-dimensional Gaussian distribution defined in eq. (3.17). Provided that the spatial correlation function appearing in the expression for Γ in eq. (3.16) is known in analytical form and

utilizing the numerical integration techniques described in [Ler02], approximations for the mean and variance integrals in eq. (3.18) can be obtained.

As a final remark, note that this analysis can give us an initial indication of or even provide elaborate bounds on the distortion performance of any spatial sampling scheme based on the *PJiG* rule. An example would be to characterize sensor subsets derived from hexagonal jittered grids instead of the squared ones discussed here. Although preliminary numerical results thereon have been presented [Lia07b], the extensive numerical investigation of eq. (3.15) via eq. (3.18) is beyond the scope of this dissertation and will not be further pursued.

3.6 Random Variable Greedy (*RaVaG*) Algorithm

A drawback of *JiG* sampling is that it is better suited for spatially stationary processes, since virtual grids are regular. However, spatial non-stationarities are in fact commonly encountered in real life physical phenomena and corresponding measurements. For example, at locations closer to windows or heat sources, temperature readings may vary more rapidly for proximate sensors than they do at dark or isolated locations, e.g. under desks. Another drawback is that it does not provide fine-grained control over the size of subsets obtained. For example, if subsets of $13^2 = 169$ sensors are inadequate, the next available choice is subsets of $14^2 = 196$ sensors. This effect becomes more pronounced for large n.

We present now a class of algorithms that aims exactly to address these shortcomings of *JiG* sampling. It is based on the expressive power of the Hilbert space H_S of sensors as random variables. The key idea of the approach is essentially

quantifying ‘colinearity’ or ‘orthogonality’ between a given candidate vector and an existing set of vectors. This can be done by using orthogonal projection error as the score function. If the orthogonal projection error of a candidate vector onto a subset of existing vectors is maximal among all such vectors, then we know that the descriptive power of the subset will maximally grow if we add the candidate to it. If, on the other hand, the orthogonal projection error of a member of the set onto the rest of the vectors in the set is minimal, then we know that the descriptive power of the set will only marginally be affected if we remove the candidate vector from it. The main strength of this approach is that it readily provides a characterization of how redundant an individual sensor is with respect to any subset of sensors for interpolation purposes, which was one of our initial goals.

Based on these concepts, our *RaVaG* algorithms for finding multiple subsets of sensors resulting in adequate interpolation (or, equivalently, multiple approximate bases for the primary subspace) proceed as follows. It is not known a priori how many subsets can be possibly created. Instead, the algorithms start creating the first subset by selecting vectors until the distortion criterion is met. Next, the second subset will be selected from the remaining vectors, and so forth. Consider, in general, a situation where we are in the process of creating the j^{th} subset. At this point, the primary subspace can be considered as being partitioned in three subspaces: 1) the space H_U of vectors in subsets 1 through j-1; 2) the space H_A of vectors already selected in subset j; 3) the space H_R of vectors not yet selected for any of the subsets. Our algorithms consider all candidate vectors η from those not yet belonging to any subset. For each

one of them, they compute the error by orthogonal projection in both of the spaces H_A and H_R (always excluding the vector η):

$$E_A(\eta) = \min \left\| \eta - \sum_p c_p \cdot \xi_p \right\|^2 \quad H_A = span\{\xi_p\}$$

$$E_R(\eta) = \min \left\| \eta - \sum_q d_q \cdot \zeta_q \right\|^2 \quad H_R = span\{\zeta_q\} \tag{3.19}$$

The important decision at this point is how to use these orthogonal projection errors to populate the subsets with 'good' vectors. A simple first choice would be to sequentially select the vector η that maximally expands the current subset, i.e., the vector maximizing $E_A(\eta)$. However, this choice does not always lead to a good subset of vectors, especially in cases where the random process is spatially non-stationary. To see this, consider a random process showing rapid variations over a small region of the field, while being smooth over the rest of the field. Then a heuristic based on maximizing $E_A(\eta)$ would first choose vectors in the rapidly varying region, because they are likely to be the most orthogonal to each other. This strategy has an immediate drawback however: it is possible that subsets subsequently constructed cannot contain any of these vectors describing rapid variations, because they have all been used up. Eventually, they will not be able to achieve the target distortion or will need to employ a much larger number of sensors.

Based on this example, there are two competing effects both of which should be taken into consideration when designing a greedy approach: expanding the expressive capability of the subset currently being constructed and not 'crippling' the expressive capability of the subset of sensors that remain. The first effect can be quantified by

requiring a high orthogonal projection error onto the space of already selected sensors, i.e., the candidate vector should be as orthogonal to this set as possible. The second effect can be quantified by requiring a low orthogonal projection error onto the space of remaining vectors, i.e., the candidate vector should be as colinear to this set as possible.

A score function can be used to express the relationship between these two effects for each candidate vector. Define a score function *Y(η)* by:

$$Y(\eta) = f(E_A(\eta), E_R(\eta)) \tag{3.20}$$

The vector η with the *maximum* score amongst all candidates is then added to the j^{th} subset. The particular choice of score function leads to different heuristics. In this dissertation we have experimented and present results with the following three score functions:

$$Y_1(\eta) = E_A(\eta)$$

$$Y_2(\eta) = E_A(\eta) - E_R(\eta)$$

$$Y_3(\eta) = E_A(\eta) / E_R(\eta) \tag{3.21}$$

The detailed subset selection algorithm is presented in Figure 3.5. The heart of the algorithm can be found at Line 12, where the score function of eq. (3.20) is computed over all candidate vectors. Computation is based on eq. (3.19) and (2.13). It can be seen that only inner products between specific sensors are needed, as opposed to possessing complete covariance information (see eq. (3.3)), and these are readily

available in the approximate Grammian $\hat{G}_0$ (see eq. (3.9)). When the algorithm terminates, there may be some remaining sensors that were not assigned to any active subsets (since they could not form a subset by themselves that satisfies the distortion target). In this case, they are distributed in a round robin fashion among existing subsets in such a way that each subset is assigned the sensor which maximizes the heuristic expression which was used for it in the first place.

1 **Input**: vectors $\{S(\boldsymbol{X}_0^p)\}_{p=1\ldots N}$, reference surfaces $\{\hat{S}_0(\boldsymbol{x},t_q)\}_{q=1\ldots W}$, Grammian $\hat{G}_{qp}$, D

2 **Output**: subsets of vectors $\boldsymbol{m}_k$

3 **begin**

4 $\boldsymbol{m}_0 \leftarrow$ I[$\bigcup_{p=1}^{N} S(\boldsymbol{X}_0^p)$], $k = 1$

5 **repeat**

6 $\boldsymbol{m}_k \leftarrow \varnothing$

7 **repeat**

8 **if** ($\boldsymbol{m}_k == \varnothing$) select first available vector into $\boldsymbol{m}_k$

9 **else**

10 **foreach** candidate vector $S(\boldsymbol{X})$

11 $p_0 \leftarrow$ I[$S(\boldsymbol{X})$]

12 $$S(\boldsymbol{X}^*) \leftarrow \arg \max_{S(X)\in \boldsymbol{m}_\theta} Y\left(\hat{G}_{p_0p_0} - \frac{\left(\sum_{p\in \boldsymbol{m}_k} (\hat{G}_{p_0p})^2\right)^2}{\sum_{p\in \boldsymbol{m}_k}\sum_{q\in \boldsymbol{m}_k} \hat{G}_{p_0p}\cdot \hat{G}_{p_0q}\cdot \hat{G}_{pq}}, \hat{G}_{p_0p_0} - \frac{\left(\sum_{\substack{p\in \boldsymbol{m}_0 \\ p\neq p_0}} (\hat{G}_{p_0p})^2\right)^2}{\sum_{\substack{p\in \boldsymbol{m}_0 \\ p\neq p_0}}\sum_{\substack{q\in \boldsymbol{m}_0 \\ q\neq p_0}} \hat{G}_{p_0p}\cdot \hat{G}_{p_0q}\cdot \hat{G}_{pq}}\right)$$

13

14 $\boldsymbol{m}_0 \leftarrow \boldsymbol{m}_0 \setminus$ I[$S(\boldsymbol{X}^*)$]

15 **until** ($\hat{E}[D_{\boldsymbol{m}_k}] \le D$)

16 $k \leftarrow k + 1$

17 **until** (($\boldsymbol{m}_0 == \varnothing$) **or** ($\hat{E}[D_{\boldsymbol{m}_0}] > D$))

18 **end**

Note: I[·] denotes the index of the argument in $\bigcup_{p=1}^{N} S(\boldsymbol{X}_0^p)$

Figure 3.5: Random Variable Greedy (*RaVaG*) subset construction algorithm

In Section 3.9, it will be shown through extensive simulations that the *RaVaG* heuristics with score functions $Y_2(\cdot)$ and $Y_3(\cdot)$ generally outperform other schemes proposed here with respect to the spatially averaged distortion criterion (3.6).

3.7 Matrix Greedy (*MaG*) Algorithm

Yet, an alternate approach to sensor selection for the spatially averaged distortion criterion (3.6) can be envisaged, owing to the simultaneous sparse approximation method described in [Cot05], [Tro06]. The latter will be hereafter referred to as Simultaneous Orthogonal Matching Pursuit (SOMP). The *MaG* algorithm builds upon manipulation of the centered data matrix, i.e., the data matrix without a spatial trend $B_0 - \hat{\mu}_0$ (see eq. (3.8) and (3.9)) and is novel for our problem.

We now describe our *MaG* algorithm. The main ideas of SOMP are also summarized here for completeness. The goal is to approximate a target signal matrix T that consists of L K-dimensional column vectors using only a few of the M column vectors $\{\varphi_\omega\}_{\omega = 1...M}$ that comprise a dictionary matrix Φ. The dictionary is typically such that M is very large. The analogy with our setting is straightforward, if we think of the collection of time series available from the sensors, as the dictionary, i.e., $B - \mu = \Phi$. The dimensionality of the columns is the length of the learning phase Θ, i.e., $K = \Theta$, while the total number of columns is the size of the network, i.e., $M = N$. A first important difference in our case however is that the signal to be approximated is the data matrix as well, i.e., $L = M = N$, and the target signal coincides with the dictionary. This creates small technical differences when actually executing the algorithm,

compared to [Tro06]. The modified algorithm, which we will refer to as simultaneous sparse approximation, is summarized in Figure 3.6, in form similar to that of [Tro06].

1 Input: Θ x N data matrix $T = B_0 - \hat{\mu}_0$, target distortion D.
Output: Sets of column indices Λ_κ.
2 Initialize the set counter $\kappa = 0$, $\Pi_0 = \varnothing$.
3 If $\kappa = 0$ initialize the residual matrix by $R_\kappa = T$. Else set $\Pi_\kappa = \Pi_{\kappa-1} \cup \{\Lambda_{\kappa-1,\, t-1}\}$ and initialize the residual matrix by $R_\kappa = T \setminus \{\Pi_\kappa\}$.
4 Initialize the index set $\Lambda_{\kappa,\, 0} = \varnothing$, the iteration counter $t = 1$.
5 Find an index λ_t that solves the (easy) optimization problem:

$$\max_{\omega \in N} \sum_{n=1}^{N} \left| < R_{\kappa,t-1} \cdot e_n, \varphi_\omega >_{\Re^\Theta} \right|$$

where e_n is the n-th canonical basis vector in $\Re^\Theta$ and $< \cdot, \cdot >_{\Re^\Theta}$ the (Euclidean) inner product in $\Re^\Theta$.
6 Set $\Lambda_{\kappa,\, t} = \Lambda_{\kappa,\, t-1} \cup \{\lambda_t\}$
7 Find the projection matrix $P_{\kappa,\, t}$ onto the span of the atoms appearing in $\Lambda_{\kappa,\, t}$.
8 Calculate the new residual:

$$R_{\kappa,t} = R_{\kappa,t-1} - P_{\kappa,t} \cdot R_{\kappa,t-1}$$

9 Increment t. Return to step 4, unless the current set meets the target distortion, as computed by eq. (3.6).
10 Increment κ.
11 If no more sensors remain or remaining sensors cannot meet the target distortion, terminate. Else return to step 3.

Figure 3.6: Matrix Greedy (*MaG*) subset construction algorithm

The second important difference from SOMP is that multiple subsets are constructed instead of a single one. In the above code, this is denoted by κ, which is the counter for subsets devised so far, while t is the counter for sensors comprising a particular subset. The key step of the algorithm can be found at Line 5, where the next column is chosen by computing its aggregate Euclidean product with all other columns. Essentially, this appoints the column that is most collinear with all other

columns by means of the Euclidean inner product. Additionally, before beginning the construction of a new subset, the residual matrix R_κ contains exactly those columns of T that correspond to sensors not yet selected in any subset. Another logical choice would be for R_κ to contain all columns of T anew, since each subset must be able to approximate the whole dataset (Line 3). This choice however, results in inferior performance in terms of devised subsets and will not be examined further.

The whole algorithm can be thought of as approximating the infinite dimensional random variables that are elements of H_S with finite dimensional (in fact, Θ-dimensional) real vectors. The computation will then lie in the Hilbert space of Euclidean Θ-vectors, thus justifying the name of the method. We expect that the actual performance of this algorithm will depend on how well the finite length vectors describe the random variables, i.e., on the length of the learning phase Θ. In the experiments' section this issue will in fact be revealed to be the Achilles' heel of *MaG*.

3.8 Greedy Potentials (*GreePo*) Algorithm for Spatially Maximum Distortion

Heuristics discussed so far pertain to the spatially averaged MSE criterion (3.6). On the other hand, criterion (3.7) for spatially maximum MSE requires a more specialized treatment, exactly because aforementioned heuristics do not explicitly take into account worst case error when choosing sensors. To better understand this we can consider the example of the *RaVaG* algorithm for devising subsets. For ease of exposition, we rewrite distortion criteria (3.6) and (3.7) for interpolation within the

primary subspace based on a discrete set of locations. The expressions are then reduced to discrete sums and maxima over discrete sets:

$$\underset{\boldsymbol{x}\in F}{avg}(D_{\boldsymbol{m}_k}) \approx \frac{1}{|A|}\sum_{x\in A} E[(\hat{S}_k(\boldsymbol{x},t)-\hat{S}_0(\boldsymbol{x},t))^2] \le D \tag{3.22}$$

$$\max_{\boldsymbol{x}\in F}(D_{\boldsymbol{m}_k}) \approx \max_{\boldsymbol{x}\in A}(E[(\hat{S}_k(\boldsymbol{x},t_i)-\hat{S}_0(\boldsymbol{x},t_i))^2]) \le D \tag{3.23}$$

In the preceding equations we have simply expressed MSEs not at continuous locations $\boldsymbol{x}$, but at a discrete set A of locations of interest, also referred to as 'test' locations. Intuitively, for any subset of vectors devised by *RaVaG*, criterion (3.22) performs *averaging* of MSEs. This effectively mitigates the effects of possible extremely bad MSEs at some locations in A. Criterion (3.23) is however based solely on the worst possible MSE. The reason heuristics such as *RaVaG* can and do fall short in this case, is that they do not proactively couple each individual sensor selection with the 'worst' regions of the observation field in terms of MSE. Intuitively, a sensor selection scheme is needed that keeps track of these sensitive regions throughout the whole selection process. Such a scheme is presented in this section.

The optimal solution to the min-max sensing topology management problem is still exponentially hard in the size of the network. It requires enumeration of all possible subsets of sensors $\boldsymbol{m_k}$ and, for each of them, computation of the maximum distortion over all test locations A according to eq. (3.23). Related problems are provably NP-hard and furthermore admit only approximation algorithms, unless P = NP [Kra07a]. Therefore, in this section we provide an approximate procedure for the min-max problem.

Recall from eq. (2.12) that the MMSE of the best linear interpolator with a subset of sensors $\boldsymbol{m}_k$ at a point $\boldsymbol{x}$ can be written as a function of the Grammian matrix G_k and the covariance vector $c_k(\boldsymbol{x})$. We repeat the expression here for convenience:

$$MMSE(\boldsymbol{m}_k, \boldsymbol{x}) = Var(\boldsymbol{x}) - \boldsymbol{c}_k^T(\boldsymbol{x}) \cdot G_k^{-1} \cdot \boldsymbol{c}_k(\boldsymbol{x}) \tag{3.24}$$

The first term thereof is the variance of the physical phenomenon at location $\boldsymbol{x}$, while the second term is the *variance reduction* at location $\boldsymbol{x}$ achieved by interpolating with subset $\boldsymbol{m}_k$. Both the variance and variance reduction are quadratic forms and thus non-negative. Since the variance depends only on the location $\boldsymbol{x}$, a possible way to find a 'good' subset from eq. (3.24) would be to require that *the minimum variance reduction for all* $\boldsymbol{x}$ *is maximized*. This gives rise to the following inequality:

$$\max_{\boldsymbol{m}_k}(\min_{\boldsymbol{x}\in A}(\boldsymbol{c}_k^T(\boldsymbol{x}) \cdot G_k^{-1} \cdot \boldsymbol{c}_k(\boldsymbol{x}))) \geq$$

$$\max_{m_k}\left(\frac{\min_{\boldsymbol{x}\in A}(\left\|\boldsymbol{c}_k(\boldsymbol{x})\right\|_2^2)}{\lambda_{\max}(G_k)}\right) \tag{3.25}$$

where $\left\|\cdot\right\|_2$ denotes the Euclidean norm and $\lambda_{max}(\cdot)$ the largest eigenvalue of the argument. The second inequality above, stems from considering the specific covariance vectors $c_k(\boldsymbol{x})$ at test locations $\boldsymbol{x} \in A$. It follows from the Courant minimax principle for quadratic forms [Moo00], the fact that G_k^{-1} is self-adjoint (because it is the inverse of a Grammian matrix) and that its eigenvalues are the inverse of the eigenalues of G_k (this can be easily seen from diagonalizing G_k and inverting). Inequality (3.25) defines a lower bound on variance reduction. The idea of our

algorithm is to greedily maximize this lower bound. Essentially, we consider how the Euclidean norm of the covariance vector $c_k(\boldsymbol{x})$ and the maximum eigenvalue of the corresponding Grammian will change over all test locations, if we add the candidate sensor to the current subset.

1 **Input**: *vectors* $\{S(\boldsymbol{X}_0^p)\}_{p=1\ldots N}$, *locations* $\boldsymbol{X}_0$, Grammian $\hat{G}_0$, D
2 **Output**: subsets of vectors $\boldsymbol{m}_k$
3 **begin**
4 $\quad \boldsymbol{m}_0 \leftarrow \mathrm{I}[\bigcup_{k=1}^{N} S(\boldsymbol{X}_0^p)]$, $k = 1$
5 $\quad$ **repeat**
6 $\quad\quad \boldsymbol{m}_k \leftarrow \varnothing$
7 $\quad\quad$ **repeat**
8 $\quad\quad\quad$ **if** ($\boldsymbol{m}_k == \varnothing$) select first available vector into $\boldsymbol{m}_k$
9 $\quad\quad\quad$ **else**
10 $\quad\quad\quad\quad S(\boldsymbol{X}^*) \leftarrow \arg \max_{S(X) \in \boldsymbol{m}_0} \left(\frac{\min_{x \in A} \left(\left\| \hat{c}_k(\boldsymbol{x}) \right\|_2^2 \right)}{\lambda_{\max}(\hat{G}_k)} \right)$
11 $\quad\quad\quad\quad \boldsymbol{m}_k \leftarrow \boldsymbol{m}_k \cup \mathrm{I}[S(\boldsymbol{X}^*)]$
12 $\quad\quad\quad\quad \boldsymbol{m}_0 \leftarrow \boldsymbol{m}_0 \setminus \mathrm{I}[S(\boldsymbol{X}^*)]$
13 $\quad\quad$ **until** $\left(\max_{\boldsymbol{x} \in A} (Var(\boldsymbol{x}) - \hat{\boldsymbol{c}}_k^T(\boldsymbol{x}) \cdot \hat{G}_k^{-1} \cdot \hat{\boldsymbol{c}}_k(\boldsymbol{x})) \le D \right)$
14 $\quad\quad k \leftarrow k + 1$
15 $\quad$ **until** $((\boldsymbol{m}_0 == \varnothing)$ **or**
$\quad\quad \left(\max_{\boldsymbol{x} \in A} (Var(\boldsymbol{x}) - \hat{\boldsymbol{c}}_k^T(\boldsymbol{x}) \cdot \hat{G}_k^{-1} \cdot \hat{\boldsymbol{c}}_k(\boldsymbol{x})) > D \right))$
16 **end**
Note: I[·] denotes the index of the argument in $\bigcup_{p=1}^{N} S(\boldsymbol{X}_0^p)$

Figure 3.7: Greedy Potentials (*GreePo*) subset construction algorithm

The actual algorithm for subset selection is shown in Figure 3.7. One candidate sensor is selected at a time as shown in Line 10, in a manner resembling the lower bound of eq. (3.25). For each candidate sensor, two quantities must be computed: the minimum norm of the covariance vector $\left\| \hat{c}_k(\boldsymbol{x}) \right\|_2^2$ over all test locations, considering

previously selected sensors along with the candidate sensor, as well as the maximum eigenvalue of the Grammian $\hat{G}_k$.

Firstly, recall from eq. (3.9) that a finite dimensional approximation of the Grammian $\hat{G}_0$ can be computed using the matrix B_0, which consists of all sensor measurements gathered during the learning phase. Any subset of sensors $\boldsymbol{m}_k$ corresponds to a submatrix B_k of B_0 and $\hat{G}_k$ of $\hat{G}_0$ adhering to eq. (3.9). Computation of the norm of the covariance vector $\hat{c}_k(\boldsymbol{x})$ is slightly more involved. A straightforward method for it would be to first obtain $W = \Theta$ reference surfaces, utilizing data from the learning phase, then extract the interpolated time series at the test locations and explicitly estimate each element of the covariance vector. This is can be done through the Euclidean inner products between the time series at the test location and the time series measured at each sensor. Denoting the columns of the submatrix B_k with $\boldsymbol{b}_1$, …, $\boldsymbol{b}_N$ and the time series obtained from reference surfaces at location $\boldsymbol{x}$ with $\boldsymbol{r}$, an estimate of the covariance vector will be given by:

$$\hat{\boldsymbol{c}}_k(\boldsymbol{x}) = \frac{1}{\Theta - 1}\begin{bmatrix} \boldsymbol{b}_1^T \cdot \boldsymbol{r} \\ \dots \\ \boldsymbol{b}_{|\boldsymbol{m}_k|}^T \cdot \boldsymbol{r} \end{bmatrix} \tag{3.26}$$

A drawback of this method however, is the computational burden of actually obtaining Θ reference surfaces through interpolation, especially in the case of a large network. A possible option to circumvent it altogether is to consider A as the collection $\{\boldsymbol{X}_0^p\}$, p = 1…N of all initial sensor positions. This is a plausible approximation for practical purposes, if the network is considered to be dense. In this case, the *GreePo* selection algorithm will essentially try to select a subset $\boldsymbol{m}_k$ so as to

minimize the maximum MSE over the locations of sensors *not in the subset*. Because $\boldsymbol{x} \in \{\boldsymbol{X}_0^p\}$, p = 1…*N*, $\hat{\boldsymbol{c}}_k(\boldsymbol{x})$ then consists only of elements in the empirical covariance matrix $\hat{G}_0$ and can be readily computed.

The quantity $\|\hat{c}_k(\boldsymbol{x})\|_2^2$, can be understood as an *information potential*, in the sense that it reveals regions of the space that bear a lot of sensors in the current subset (high potential) as well as regions of the space that are devoid of sensors (low potential). Therefore, the whole algorithm is referred to as the Greedy Potentials algorithm (*GreePo*). Upon termination, there may remain some sensors that were not assigned to any subsets (since they could not form a subset by themselves satisfying the distortion target). These are distributed randomly in a round robin fashion among existing subsets.

3.9 Evaluation

3.9.1 Spatially Averaged Distortion

We first tested our sensing topology management schemes with a range of synthetic data. A purely simulated evaluation setting has the major advantage that it gives us access to the ground truth, i.e., the spatial process itself. As a result, we can evaluate the true distortion of interpolation with a subset of sensors as compared to the realizations of the monitored process.

The setting for experiments was an observation field of square shape and size 10^4 m^2. We considered Poisson-based random deployments with $N = 500$, $N = 1000$, $N = 1500$ sensors. This serves to evaluate the impact of the level of initial over-

deployment on our algorithms of Figures 3.3, 3.5 and 3.6. Recall also that our algorithms require a learning phase of Θ time instants. Readings obtained during the learning phase are used for two purposes:

- Construction of W reference surfaces $\hat{S}_0(\boldsymbol{x}, t_q)$ for distortion estimation according to eq. (3.6). Distortion estimation is required by all subset construction algorithms presented in this chapter.
- Construction of the data matrix B_0 and possibly estimation of the covariance matrix $\hat{G}_0$ through eq. (3.8) and (3.9) respectively. This is required only by the *MaG* and *RaVaG* algorithms.

Distortion estimation is a computationally expensive operation and must be repeated after each iteration that adds a sensor to the current subset (step 4 in Figure 3.3, step 15 in Figure 3.5 and step 11 in Figure 3.6), so a lower W is preferable. On the other hand, the orthogonal projection error of eq. (2.13) which is used in *RaVaG* by virtue of eq. (3.21), assumes linearly independent vectors and the covariance matrix estimate of eq. (3.9) should therefore be full rank [Moo00]. This translates into $\Theta \geq N$. In principle, a higher Θ also enables better estimation performance through potential comparison with more reference maps. The compromise we followed in our simulation results is to initially run the learning phase for $\Theta = N$ time instants and then use all acquired sensor values for covariance matrix estimation, but only $W = 125$ reference maps for distortion estimation. In practice, we found that even the requirement $\Theta \geq N$ is not essential to obtain good subsets. We will show that performance of the *RaVaG* algorithms does not significantly degrade even for $\Theta \approx N / 4$. Data interpolation was performed with two methods: reference maps were

constructed with least squares spline interpolation to provide robustness against noise. In all other occasions, i.e., during subset construction and performance evaluation of the devised subsets, a standard, and much faster, interpolation method based on Delaunay triangulation was used.

There is no existing solution to actually compare our scheme against. However we have devised and experimented with a reasonable alternate approach. The basic premise is to construct subsets by choosing sensors at random. Specifically:

Random selection: Select k sensors at random to comprise a subset. Keep increasing k until the distortion criterion (3.6) is satisfied. Repeat until no sensors remain unselected or are too few to meet the target distortion.

3.9.1.1 Stationary Processes

We first conducted experiments with a spatially stationary physical process. Process realizations were generated according to a *simple kriging* model [Ste99], which is commonly used in geostatistics and atmospheric sciences to describe environmental data. Specifically, zero mean white noise was fed into a symmetric 2-D low pass spatial filter. The target distortion was set to 0.5. White Gaussian measurement noise of mean zero was added to sensor samples in all cases, resulting in a Signal to Noise Ratio equal to 10. It is worth mentioning that another well known model for generation of spatially correlated sensor network data [Jin06] could not be used in our case: for our evaluation, the ground truth needs to be continuous over the

entire observation area whereas this model provides data values only at specific sensor locations.

Figure 3.8(a) shows the number of subsets obtained with random selection as well as the *JiG*, *MaG* and *RaVaG* algorithms with all different score functions (see eq. (3.21)), for $N = 500$, $N = 1000$ and $N = 1500$. Figure 3.8(b) shows the average size of subsets for each scheme, along with standard deviations. Sizes are the ones obtained immediately after running our selection algorithms, i.e., before assignment of remaining sensors among devised subsets. The different network sizes serve to indicate the impact of initial over-deployment on the results.

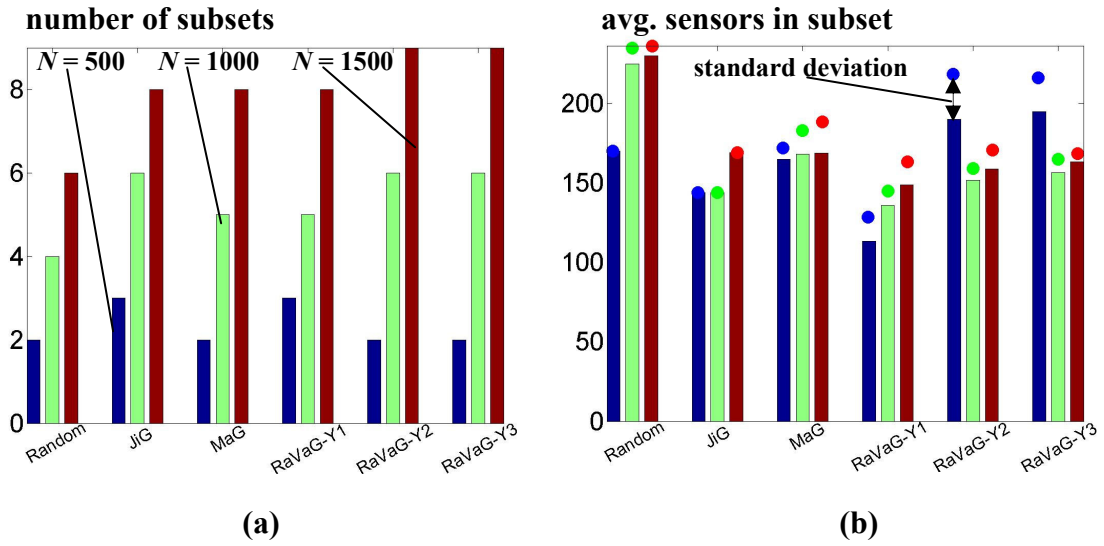

Figure 3.8: (a) Numbers of subsets devised for $N = 500$, 1000, 1500 and stationary data (b) Average sizes of subsets with standard deviations

In general, efficiency is improved with more available subsets or equivalently with smaller subset sizes. It can be seen that our schemes greatly improve on the efficiency of an unscheduled network (consisting of only one set) by 3, 6 and 9 times for $N = 500$, $N = 1000$ and $N = 1500$ respectively. The schemes that exhibit the best

performance in this respect are *RaVaG-*Y_1 for N = 500 and *RaVaG-*Y_2 for both N = 1000 and N = 1500. Additionally, improvement on the number of subsets is also achieved in comparison to random selection, specifically by 50% for all values of N = 500, N = 1000 and N =1500. This improvement can be explained by observing average subset sizes. For example, the average subset devised with *RaVaG-*Y_2 for N = 1000 consists of 151 sensors. This is a 33% reduction over the 225 sensors comprising the average random subset for the same value of N.

Figure 3.9 plots instantaneous square error versus ground truth for one run and for each value of N. The schemes compared are random selection and *RaVaG* with score functions $Y_1(\cdot)$ and $Y_2(\cdot)$ which showed the best performance in terms of number of devised subsets in Figure 3.8(a). The subsets were activated in a sequential manner, with vertical lines indicating points of switching between subsets and numbers on top indicating the sequence number of the subset. The ensemble mean in time of the spatially averaged squared error is also shown in the figures as '*E.M.*'. Subsets used for these experiments were the finalized ones, i.e., the ones obtained with distribution of residual sensors after completion of the algorithms, as would happen in a real setting. In a real setting, the sensor network would also be active for periods of time much longer than the 250 time instants shown here; Figure 3.9 merely serves to examine whether distortion achieved by the sensor subsets adheres to the target. Realizations up to 50 correspond to distortion during the learning phase, where all sensors in the network are reporting. This is done to give an idea of the performance of the initial deployment and how a low initial distortion can be effectively traded off for multiple reporting subsets.

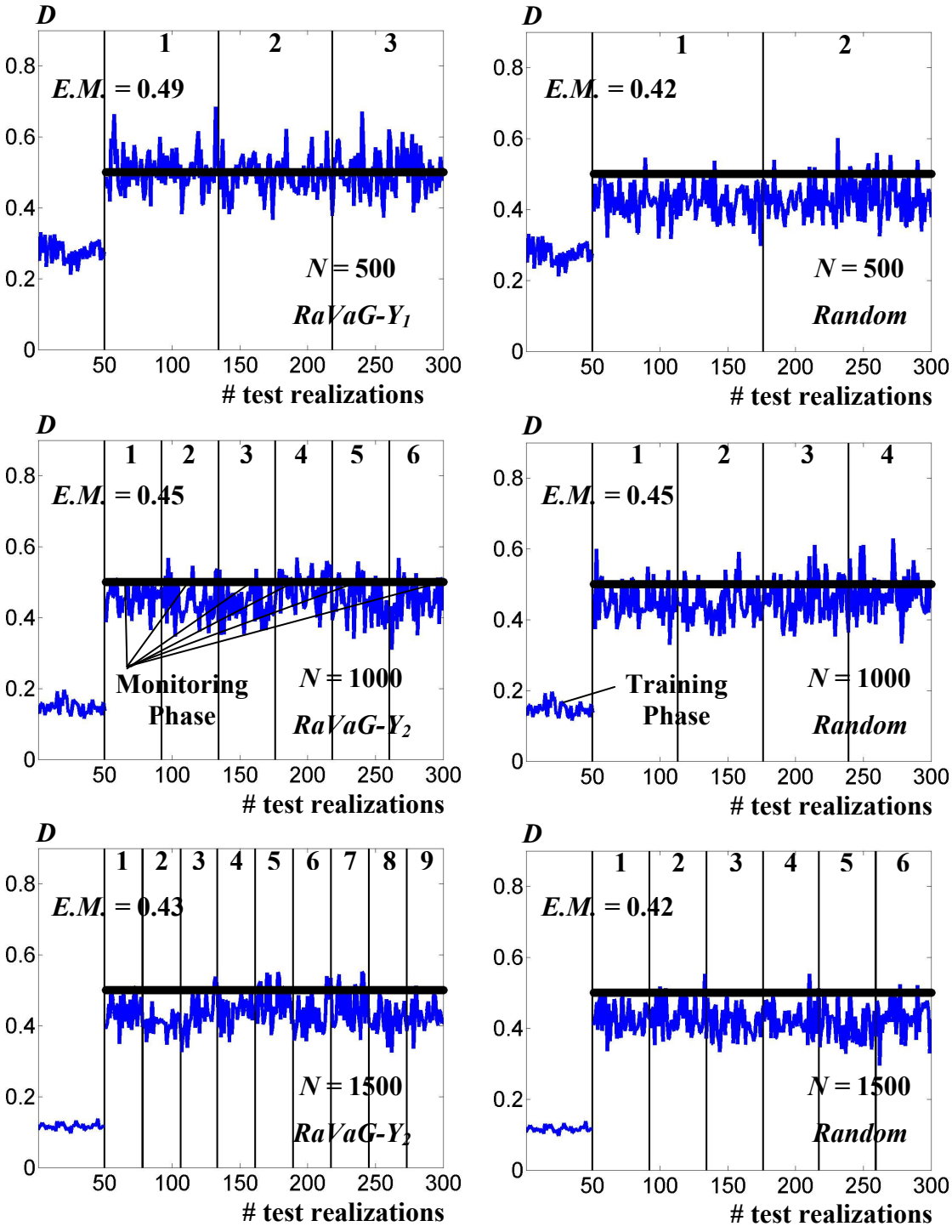

Figure 3.9: Instantaneous spatially averaged squared error vs. ground truth for a stationary process

It can be seen that subsets devised with the proposed schemes, although smaller, still succeed in providing sufficiently good interpolations of the monitored process. Most importantly, they do so in a much stronger sense than what is actually required by criterion (3.6). Specifically, eq. (3.6) asserts that the Mean Squared Error averaged over space should be less than the distortion target, and the 'mean' is taken over time. Mean Squared Error is approximated in the figures by the ensemble mean. Figure 3.9 shows however, that squared error, if averaged over space, is close to the distortion bound *for each time instant*, as opposed to just the mean sense over all time instants. This is an important aspect of performance achievable with our schemes, for a spatially averaged distortion criterion, which however depends on some conditions. These will be discussed in Section 3.9.1.3.

3.9.1.2 Non-stationary Processes

Next, we experimented with processes characterized by a linearly growing trend as well as a covariance structure changing over the observation field, i.e., not spatially invariant. To generate appropriate process realizations we followed the procedure described in [Not02] for estimation of non-stationary behavior, which has been utilized previously in sensor network literature [Gue05], [Kra06], [Kra07b]. The basic idea is to model a non-stationary process as a spatially varying linear combination of stationary processes. In our case, two such stationary processes were generated with spatial filters corresponding to Nyquist sampling rates of 100 sensors and 400 approximately. To combine them, the observation field was first divided into two regions along its main diagonal. The spatially varying weights were then chosen as

two Gaussian kernels, each centered at one of these distinct regions on the field. The contribution from each of the stationary processes becomes prominent only towards the center, i.e., the peak of the corresponding weighing kernel. A linear trend starting at the leftmost edge of the field at value 15 and reaching the rightmost edge of the field at value 35 was finally superimposed on the non-stationary covariance data. This could for instance describe a steadily growing trend in temperature as one moves towards the windows in a room. The entire procedure resulted in process realizations that would require approximately 225 randomly selected sensors to achieve a target distortion of 0.5 (same as in the stationary case).

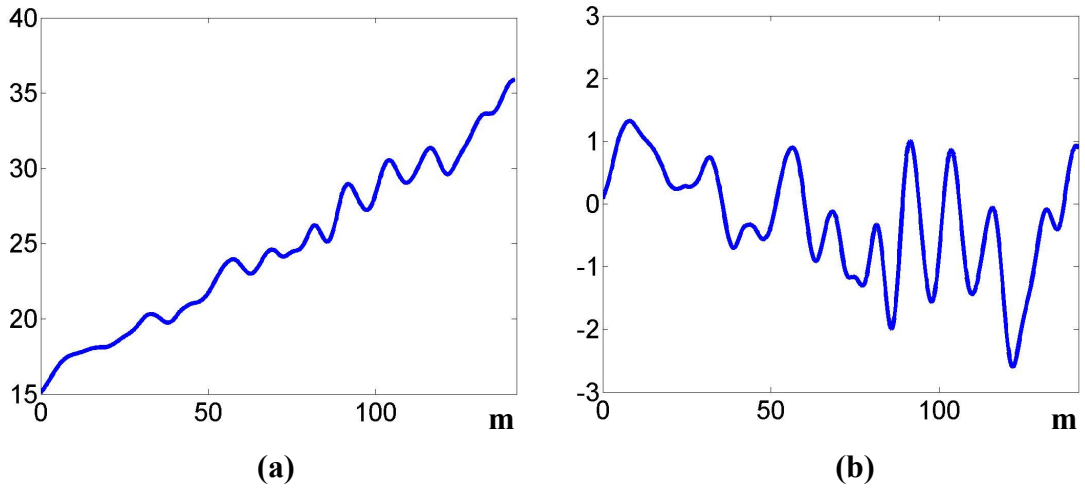

Figure 3.10: Cross section of a non-stationary realization along main diagonal of observation field shown (a) with trend and (b) after subtracting trend

Figure 3.10(a) shows a cross section of the covariance structure for one process realization taken along the main diagonal of the observation field. Measured values increase as one moves towards the upper right corner of the field, due to the linear trend. Figure 3.10(b) shows the same cross section if the trend is subtracted. It can be seen that the resulting process shows more rapid variations for the upper half of the

cross section and corresponds to the component stationary process with lower correlation.

Figures 3.11(a) and (b) respectively show the numbers of devised subsets and average subset sizes for $N = 500$, $N = 1000$ and $N = 1500$. The gains in terms of total number of subsets when compared to the unscheduled case are 3, 6 and 8 fold for $N = 500$, $N = 1000$ and $N = 1500$ respectively. This superior performance was achieved by *RaVaG-*Y_2 for $N = 500$, *JiG* for $N = 1000$ and *RaVaG-*Y_3 for $N = 1500$. In comparison to random selection a gain in the number of subsets is only achieved for the two larger values of N and amounts to 50% in both cases.

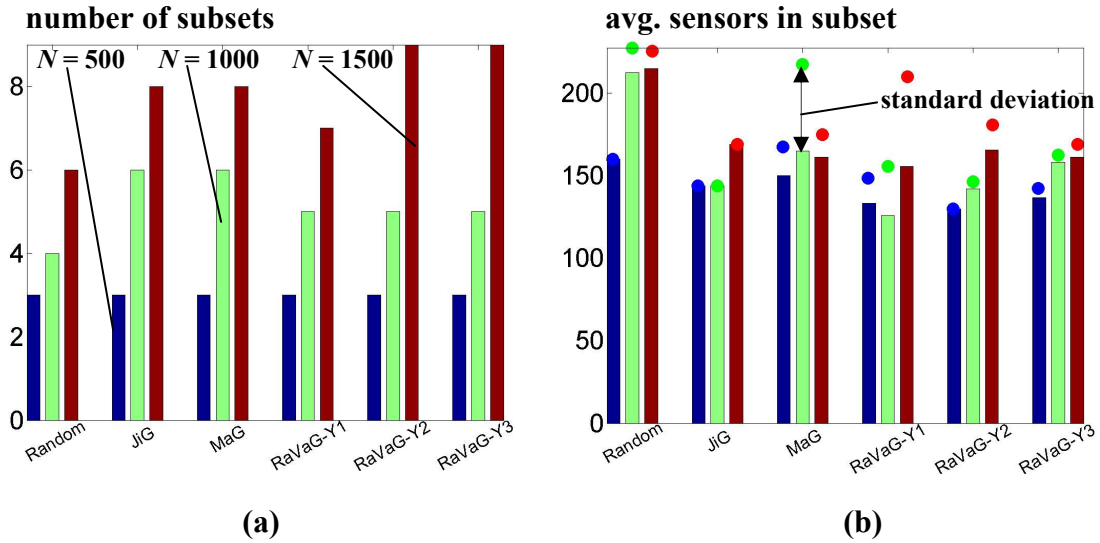

Figure 3.11: (a) Numbers of subsets devised for $N = 500$, 1000, 1500 and non-stationary data (b) Average sizes of subsets with standard deviations

Figures 3.12(a) and (b) depict a 'bird's eye' view of sensors comprising a subset for $N = 1000$, as chosen by the random scheme and *RaVaG* with cost function $Y_2(\cdot)$ respectively. Small dots represent all deployed sensors; circles represent sensors

selected by the random scheme while squares represent sensors selected by *RaVaG*. It can be observed that randomly chosen sensors often display erratic patterns, with swarms and spatial gaps. By contrast, the Hilbert space approach correctly selects more sensors to account for the higher variability of the physical phenomenon in the upper right half of the observation field (compare also to Figure 3.10).

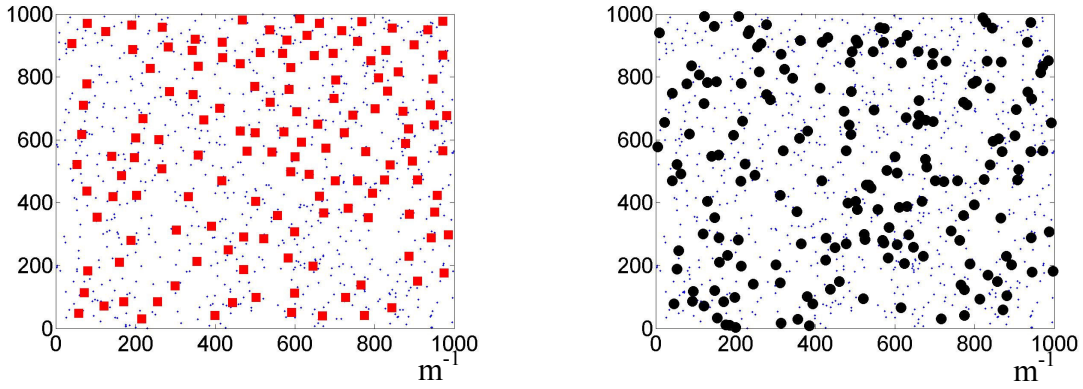

Figure 3.12: Sensors comprising a single set for $N = 1000$ for (a) *RaVaG-*Y_2 and (b) Random Selection

The instantaneous interpolation performance of the best schemes in terms of number of produced subsets is shown in Figure 3.13, alongside that of random selection. It can be seen that the distortion target is well met by our schemes in an instantaneous squared error sense.

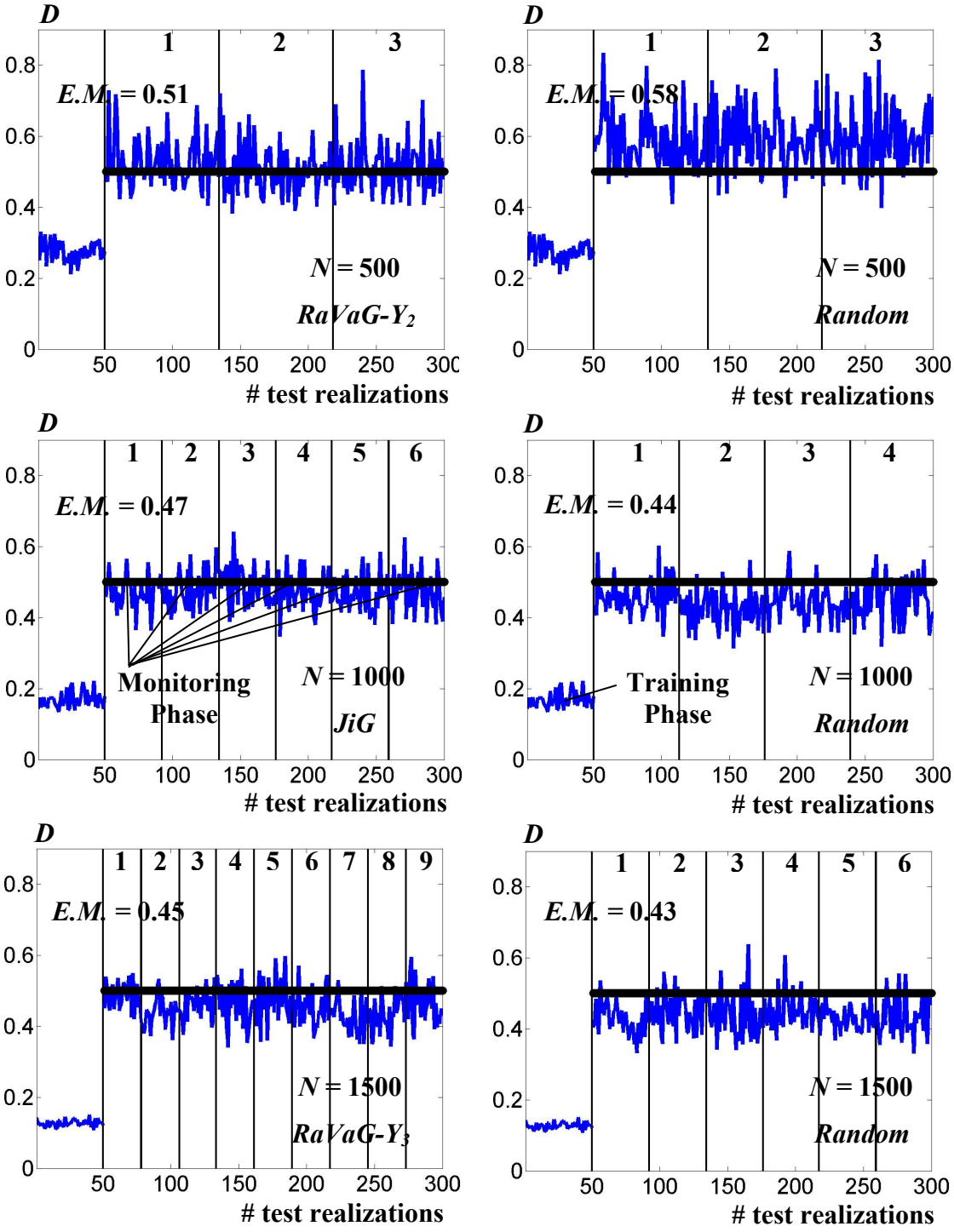

Figure 3.13: Instantaneous spatially averaged squared error vs. ground truth for a non-stationary process

3.9.1.3 Discussion

From Figure 3.13 it can be observed that subsets devised by random selection show degraded MSE performance in the case of $N = 500$. Specifically, squared error, even if averaged over time does not meet the distortion target of 0.5. The reason is that, in this case, distortion computed through eq. (3.6) is underestimating the true distortion. For proper distortion estimation, initially deployed sensors should provide a relatively accurate approximation of the ground truth to begin with. This is apparently not the case for $N = 500$, which is also evident by the fact that distortion during the learning phase is considerably close to the target distortion.

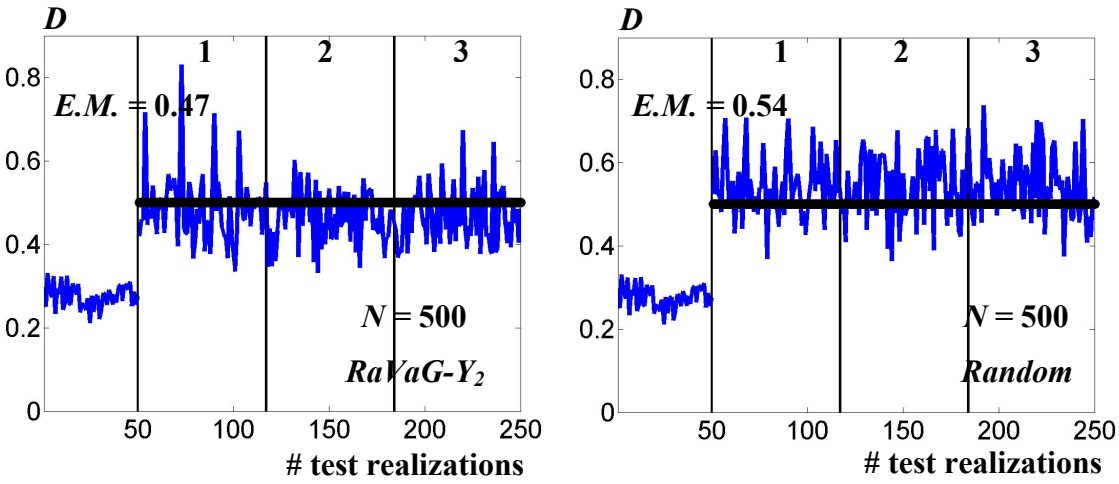

Figure 3.14: Instantaneous spatially averaged squared error vs. estimated truth for a non-stationary process

To illustrate this point further, Figure 3.14 shows the instantaneous distortion incurred by the sensor subsets selected when $N = 500$, not against the ground truth as in Figure 3.13 but against least squares spline interpolations of 200 test realizations (the same method is used to obtain the reference surfaces of eq. (3.6)). It can be seen that average distortion is now closer to the target. This shows that the discrepancy for

$N = 500$ in Figure 3.13 is due to inaccurate reference surfaces, i.e. inaccurate distortion estimation and not caused by a deficiency of the random sensor selection scheme. It also indicates that, in Figure 3.13, and for $N = 500$, the *RaVaG-Y_2* scheme, even though it does not achieve a gain in terms of additional subsets, nevertheless adequately compensates for the inaccurate distortion estimation by selecting better sensors altogether. Therefore, it still fares better than random selection.

This shortcoming of distortion estimation can be remedied by 'correcting' the distortion target fed to the sensor selection algorithm in the first place. The actual magnitude of this correction depends on the sensed spatial process and the type of deployment (e.g. Poisson) and can only be determined empirically through simulation. To give an example pertaining to Poisson deployments, let us call *estimated oversampling factor*, the ratio of N versus the estimated size of a randomly selected subset necessary to achieve the target distortion. From Figure 3.11(b) the estimated oversampling factor would be 3.1 ($\approx 500 / 160$) for $N = 500$, 4.4 ($\approx 1000 / 225$) for $N = 1000$ and 6.5 ($\approx 1500 / 230$) for $N = 1500$. The correction in the distortion target for each estimated oversampling factor can then be chosen as the difference in squared error when comparing to the ground truth and when comparing to a reference surface, averaged over time. This difference is essentially the difference (i.e. obtained by subtraction) in instantaneous distortion as shown in Figure 3.13 for $N = 500$ and the distortion target, averaged over time, and can be specifically computed to be ~0.08 for $N = 500$. Thus, for example, for Poisson deployments and estimated oversampling factor ~3.1, the correction needed will be ~0.08, i.e., the distortion target should be 0.42 instead of 0.5. An association with oversampling factors (i.e., disregarding

network sizes) is reasonable in this case, because the Poisson process is a 'completely random' point process. For other types of initial deployments however, detailed curves of initial network size vs. distortion correction have to be built in simulation. This is however an issue going beyond the scope of this dissertation.

From Figures 3.8 and 3.11, we see that obtained sizes are similar across different subsets for the same algorithm and for the same number of sensors N, as expected. For different algorithms there is a small deviation in sizes, depending on how the algorithm actually selects sensors, which is also expected. Subset sizes vary more for nonstationary data and for $N = 500$. In general, the variants of *RaVaG* more consistent in providing good performance were those utilizing score functions $Y_2(\cdot)$ and $Y_3(\cdot)$. The performance of the variant based on $Y_1(\cdot)$, even though reasonably good in the stationary case, was observed to deteriorate for non-stationary data. This is indicated by large variations in the sizes of the subsets devised by this variant (e.g. for $N = 1000$ the second subset is much smaller than the rest). Variations can be attributed to the cost function being 'fooled' into selecting many sensors in the region of the field characterized by lower correlation, which cripples further construction of subsets, as described earlier as an example in section 3.6. However, *RaVaG*-Y_1 still devises more subsets in the case of $N = 500$ and stationary data, because it expands the span of the subspace corresponding to the current subset more aggressively than *RaVaG*-Y_2 or -Y_3.

A further issue of interest is how much performance of the algorithms is actually affected by the length of the learning phase Θ. We have in fact run *RaVaG* as well as *MaG* for the case of $N = 1000$ with a much smaller learning phase than what is required to ensure linear independence in eq. (2.13), namely $\Theta = 250$. A significant

advantage of *RaVaG* is revealed then, as shown in Figure 3.15. *RaVaG* fares as well as in the case of an ideally long learning phase, i.e., $\Theta = N = 1000$ because it uses the data matrix only to estimate covariances. The *MaG* algorithm on the other hand, exhibits heavily degraded performance for non-stationary data, with the number of subsets being reduced from 6 to 4 (note that the subset sizes are still similar with the $\Theta = 1000$ case). The reason is that it tries to approximate N column vectors that are Θ – dimensional in this case. The effect can be thought of as trying to approximate a redundant dictionary (because there are N columns > Θ dimensions) with itself.

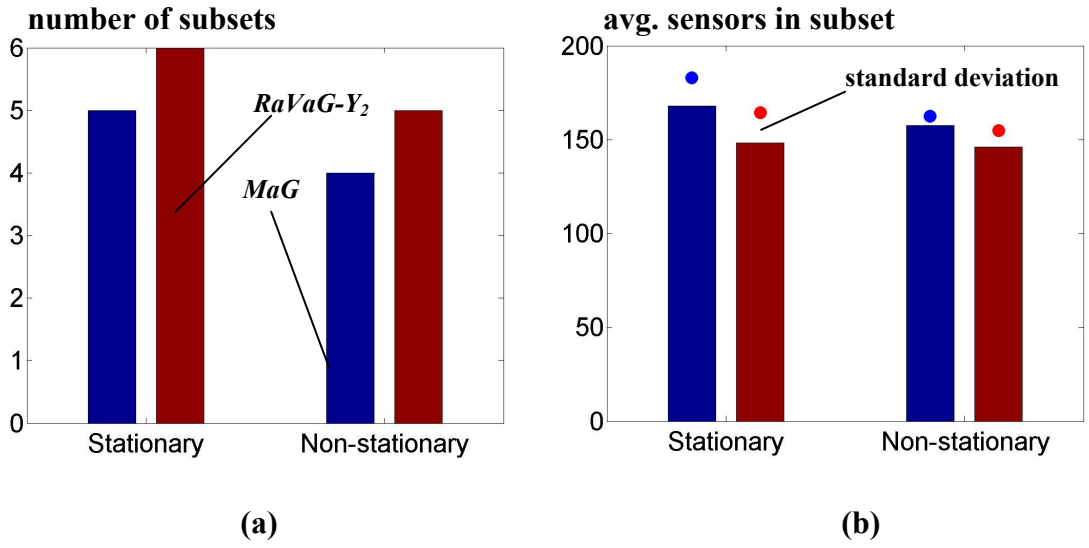

Figure 3.15: (a) Numbers of subsets devised for $N = 1000$ and $\Theta = 250$ for stationary and non-stationary data (b) Average sizes of subsets with standard deviations

Overall, the results shown in Figures 3.8-3.13, indicate that our approaches succeed in finding good disjoint subsets that meet a specified target distortion. No single one of the proposed schemes strictly outperforms others in all cases. However, the proposed *RaVaG* algorithm with score functions $Y_2(\cdot)$ or $Y_3(\cdot)$ does consistently produce good results, especially for non-stationary underlying phenomena. By having

only one of these subsets active at each point in time, desired sampling-interpolation performance can be achieved. Subsets may be activated either in a sequential manner, i.e., each subset is activated after the previous one has depleted its energy resources or in an interleaved manner to achieve better robustness. For example, the simplest such option would be to rotate control in a round robin fashion amongst subsets for consecutive time instants (this should be done intelligently nevertheless, to avoid the energy overhead of switching between subsets). In general, by using each subset for the same average percentage of time, the network can ideally remain operational for durations proportional to the total number of subsets. In the following section we will present results on the energy expenditure of our selection algorithms using a simple on-demand scheme of switching sensing control between different subsets.

3.9.1.4 Energy Expenditure

Exact energy expenditure for both an unscheduled network and one employing sensing topology management will depend on the particular radio transceivers and modes of transmission used, the message lengths, the wireless medium conditions (e.g. attenuation, fading) and the spatial reuse of this medium provided by the Medium Access Control (MAC) protocol. The contribution of all these aspects to energy consumption however, is largely dependent on the number of transmitted, received and overheard packets. This is a common protocol abstraction technique [Kin06] and the aforementioned quantities are used as energy consumption metrics for this section. For the majority of available sensor node radios, the power expended for reception is comparable with that expended for transmission [Kri05]. In our simulations therefore,

transmitted and received packets are added together. Additionally, overhearing power can either be comparable to reception power or close to zero, for example if an efficient overhearing avoidance technique is employed [Hu08]. Hence two extreme cases for overhearing would be not to count overheard packets at all and to count them as received packets.

Our sensing topology management schemes have energy overhead compared to an unscheduled network. Specifically, sensor nodes should know if they belong to the currently operating subset. Therefore, whenever a switch between monitoring subsets is necessary, the data center floods a subset membership packet into the network. This packet could be a simple bit mask with length equal to the number of nodes in the network. During flooding, each node stores identities of nodes from which it received the bit mask packet to serve as possible forwarders back to the data center. We call these nodes 'parents'. When flooding is complete, only sensors that belong to the current monitoring subset forward their data packets to a randomly selected parent, which in turn forwards them to one of its parents. In this manner, a data gathering tree is built, rooted at the data center.

Figure 3.16 shows two extreme cases of the flooding overhead in packets per node versus transmission range: when overhearing is completely avoided and when it consumes energy similar to reception. All simulations were run on three different networks, of sizes $N = 500$, $N = 1000$ and $N = 1500$ and for a single flooding operation. Transmission ranges were picked as multiples of an equivalent sensing range, along the lines of work on event detection [Zha05]. In our interpolation setting we used as equivalent sensing range the average radius of Voronoi cells centered on

the nodes picked by the *RaVaG-Y*$_2$ algorithm for a stationary process and N = 1500. This was specifically 5.2 m.

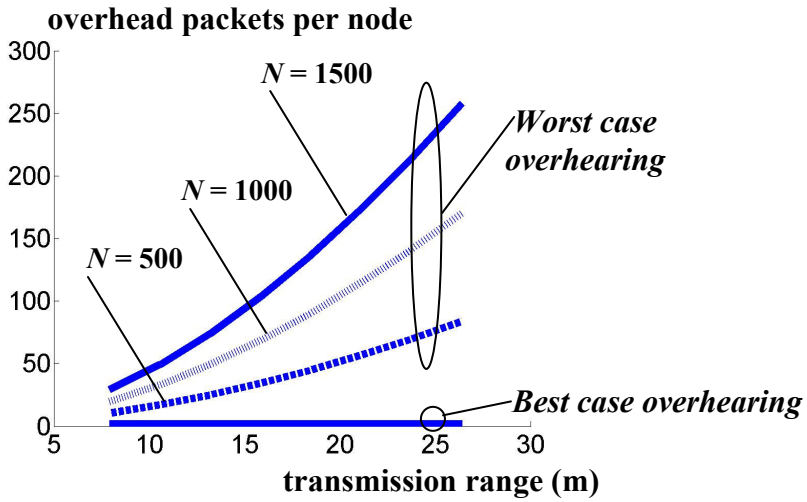

Figure 3.16: Flooding with best and worst case overhearing

To construct the curves, the data sink was assumed to be located at one of the corners of the observation field. A membership packet originating from the data sink was then flooded through the network. Upon reception of the packet, each node, broadcasts it exactly once to all of its neighbors. When overhearing can be completely avoided, a node effectively receives the packet exactly once, so the total number of transmissions and receptions per node is 2, i.e., each node receives and transmits the membership packet exactly once. When overhearing cannot be avoided, each node effectively hears the packet from a number of nodes in the order of its average number of neighbors. Since this number grows proportionally to the square of the transmission range, the overhead grows accordingly with transmission range. The quantity plotted in the end is obtained by summing all transmitted, received or virtually received, i.e., overheard, membership packets, and dividing by the total number of nodes in the

network. This is essentially a *normalized* measure for energy consumption. It can be equivalently interpreted as if every node in the network consumes energy equal to that needed for reception of the plotted amount of packets.

Figure 3.17 shows the sum of transmissions and receptions per node for a single data gathering run versus transmission range, for $N = 500$, $N = 1000$, $N = 1500$. The purpose is to compare energy expended for data gathering by the best of our proposed schemes with that of an unscheduled network. According to Figures 3.8 and 3.11, these best schemes were:

- *RaVaG-Y*$_1$ for $N = 500$, *RaVaG-Y*$_2$ for $N = 1000$ and $N = 1500$ and stationary data.
- *RaVaG-Y*$_2$ for $N = 500$, *JiG* for $N = 1000$ and *RaVaG-Y*$_3$ for $N = 1500$ and non-stationary data.

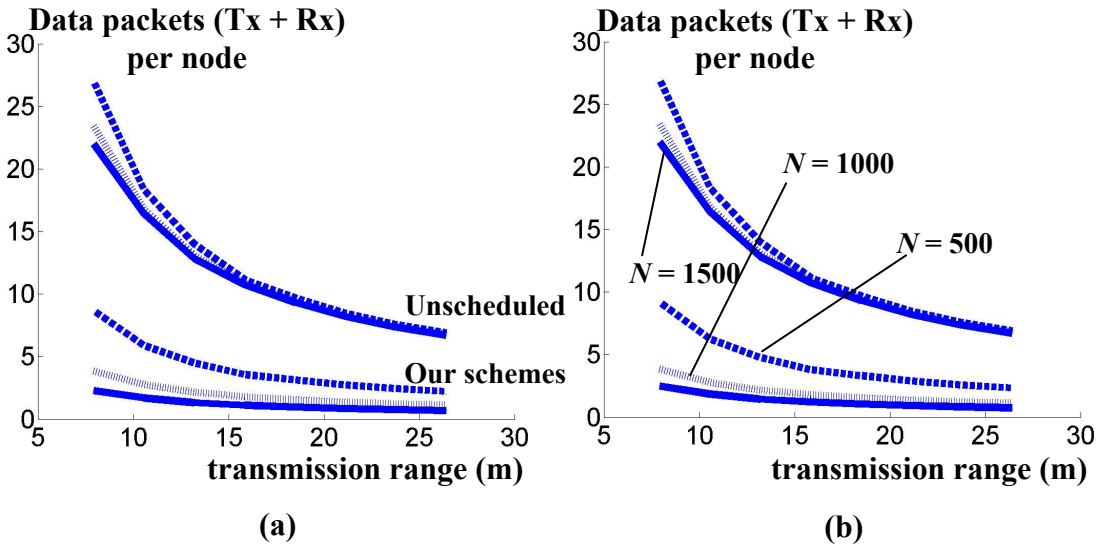

Figure 3.17: Transmissions and receptions of data packets for our schemes compared to an unscheduled network for (a) a stationary process (b) a non-stationary process

To construct the curves, data packets originating at each of the nodes comprising the subset were forwarded through randomly selected parents towards the data sink. All transmissions and receptions of data packets were added together, and then divided by the number of nodes in the network N. Only transmissions and receptions were counted, because once the data senders have been decided upon, Time Division Multiple Access (TDMA) techniques [Deg06] can be used to avoid overhearing altogether. All results are shown for the second subset devised by respective algorithms. Results are very similar for other subsets.

Energy expenditure is higher for small transmission ranges that force data packets to go through more hops towards the data center. Values are close for unscheduled networks with $N = 1000$ and $N = 1500$, because the networks are dense. Values are also close for our schemes in the stationary and non-stationary cases, because the sizes of devised subsets are similar as denoted by Figures 3.8 and 3.11. It can be seen that for all transmission ranges the energy expenditure per node is significantly reduced compared to an unscheduled network, specifically by 91%, 84% and 67% for $N = 1500$, 1000 and 500 respectively (percentages are roughly constant across transmission ranges and similar for stationary and non-stationary data).

Switching between subsets should normally happen much less frequently than sending data packets. For example, if the nodes measure temperature every ten minutes, a whole day of monitoring with a specific subset would translate to 144 rounds of data reporting. By comparing the absolute difference in the energy expenditure curves of Figure 3.17 with the overhead values in Figure 3.16 it can be seen that the overhead of our schemes quickly becomes negligible. This means that

after a short number of rounds (specifically 42 for the highest transmission range shown) of reporting e.g. with a *RaVaG* subset, we start gaining in terms of energy consumption compared to the unscheduled case. It is also stressed that the case of overhead presented in Figure 3.16 is 'worst case' in the sense that much more energy efficient flooding methods have been proposed for sensor networks [Kin06].

Figure 3.18 compares transmissions and receptions per node for one data gathering run between our schemes and random selection. Curves for our schemes are exactly the same as in Figure 3.17. Subsets thus devised, reduce the energy consumption per node compared to random selection by 37% for all values of N and stationary data. This means that, in the case of N = 1500 for example, and since plotted values are normalized, a sensor node will be able to report roughly for a factor of $1 / (1 - 0.37) = 1.58$ more time under our schemes than under random selection. For non-stationary data, the reduction is 33% for N = 1500 and 36% for N = 1000. Although the case N = 500 shows no apparent improvement, it has already been pointed out in Section 3.9.1.3 that randomly selected subsets do not actually adhere to the distortion bound here, i.e., this case is somewhat pathological.

Although our setup for examining energy consumption has been a simple one, we believe that it nevertheless captures the essence of multi-hop data gathering scenarios. From aforementioned results, we can therefore draw the conclusion that our schemes do indeed show significant merit with respect to reducing the energy expended for spatial interpolation with a WSN.

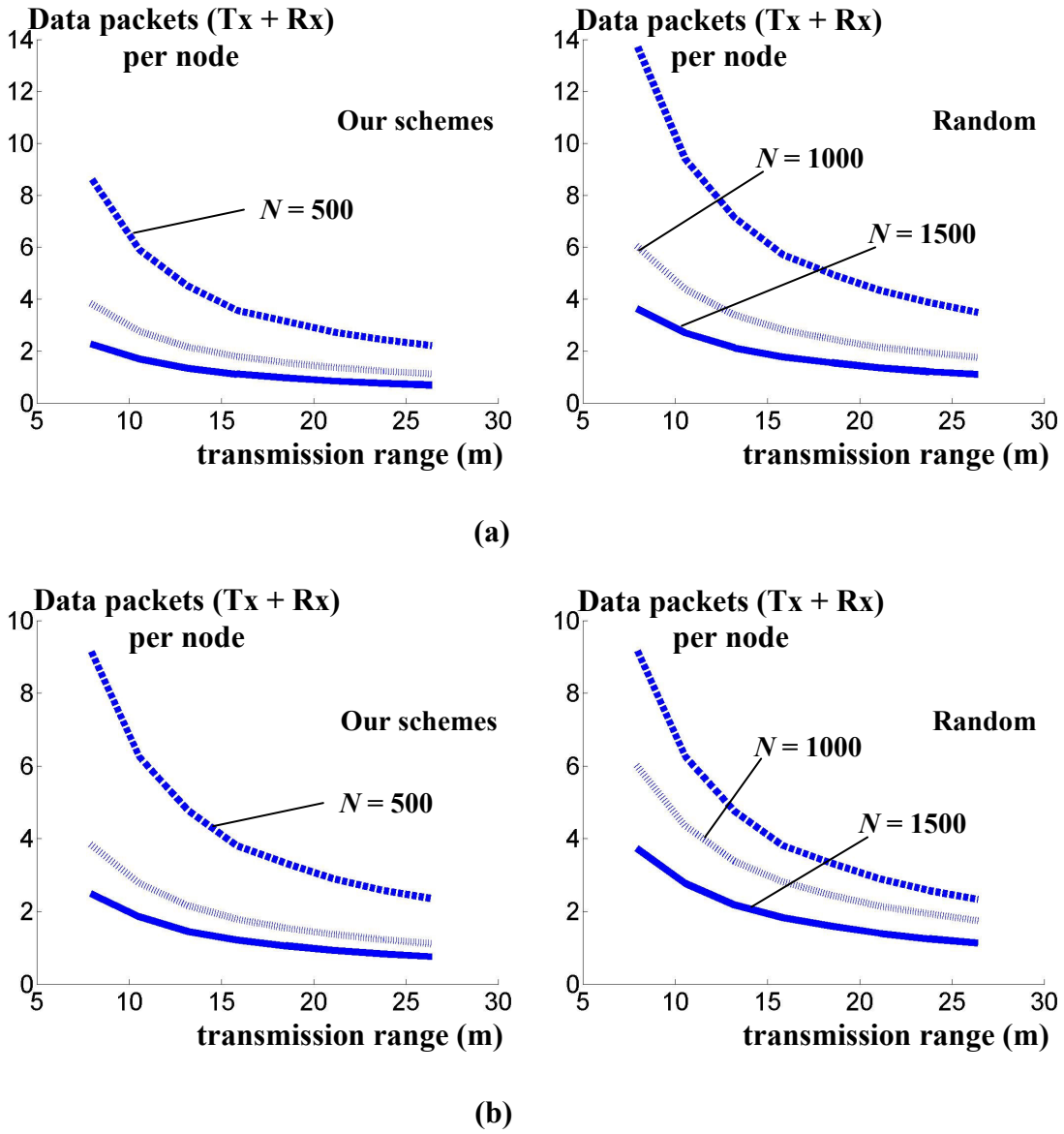

Figure 3.18: Transmissions and receptions of data packets for our schemes compared to random selection and (a) a stationary process (b) a non-stationary process

3.9.1.5 Real Data

Our final experiments were conducted on samples of ambient temperature from the LUCE Sensorscope deployment at EPFL [Sen07]. Samples were being collected by 97 Sensorscope nodes, which were dispersed over the entire EPFL campus and

reported meteorological data several times a day. For our purposes, we used 83 sensors forming the largest subset with data available for a common, sufficiently long period of observation. This observation period was from April 24th to May 9th, 2007. As a result, we had N = 83 sensors, each reporting 12 times a day for 15 days (April 29th was missing from available datasets). Half of these samples were used for the learning phase and the rest for checking instantaneous interpolation distortion.

One of the problems, intrinsic to testing on real data, is that we do not know the underlying physical phenomenon (i.e. the ground truth). The only information available is what we can learn from all reporting sensors, and we evaluated distortion as compared to this situation. This means that we compared to the primary subspace H_{X0} instead of H_S, and implicitly assumed that this is a sufficiently good approximation of the real process (i.e. we start of with sufficiently dense sampling to capture the relevant information). The target distortion for devising subsets was set to $[2\ ^0\text{F}]^2$. This represents mean squared distortion and corresponds to 2 ^{0}F of actual point distortion when predicting temperature.

For the chosen target distortion, random selection, as well as *RaVaG-Y_1* and *MaG* resulted in only one reporting subset. *JiG*, as well as *RaVaG-Y_2* and *RaVaG-Y_3* performed better however, devising two subsets of 42 and 41, 37 and 46 sensors and 32 and 51 sensors respectively. Figure 3.19 shows instantaneous performance of these subsets for the $Y_3(\cdot)$ variant. It can be seen that the target distortion is met on the average, rendering our approach a viable alternative to having all sensors reporting. Outlier values in the instantaneous distortion curve can be attributed to anomalous readings of specific sensors in combination with relative under-sampling. For

example, if a sensor sometimes happens to be near a source of excessive heat, its readings are going to deviate from those of the rest of the sensors which measure only environmental temperature. This will have a pronounced influence on interpolation for subsets that do not include this sensor, if no other sensors are available to sense the anomalies.

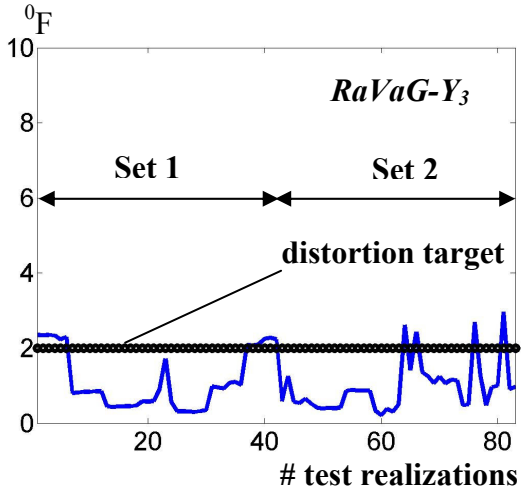

Figure 3.19: Instantaneous spatially averaged error for real temperature data

3.9.2 Spatially Maximum Distortion

We now turn our attention to the *GreePo* algorithm described in Section 3.8 for spatially maximum distortion. The simulation setup is exactly the same as in the spatially averaged distortion case, albeit only the case $N = 1500$ is examined. The reason behind this choice is that for the same distortion target of 0.5 and a Poisson type deployment, the spatially maximum distortion criterion (3.7) is much more demanding in terms of sensors needed, when compared to the spatially averaged criterion (3.6). When evaluating related sensing topology management approaches, an application to smaller networks would thus be less meaningful.

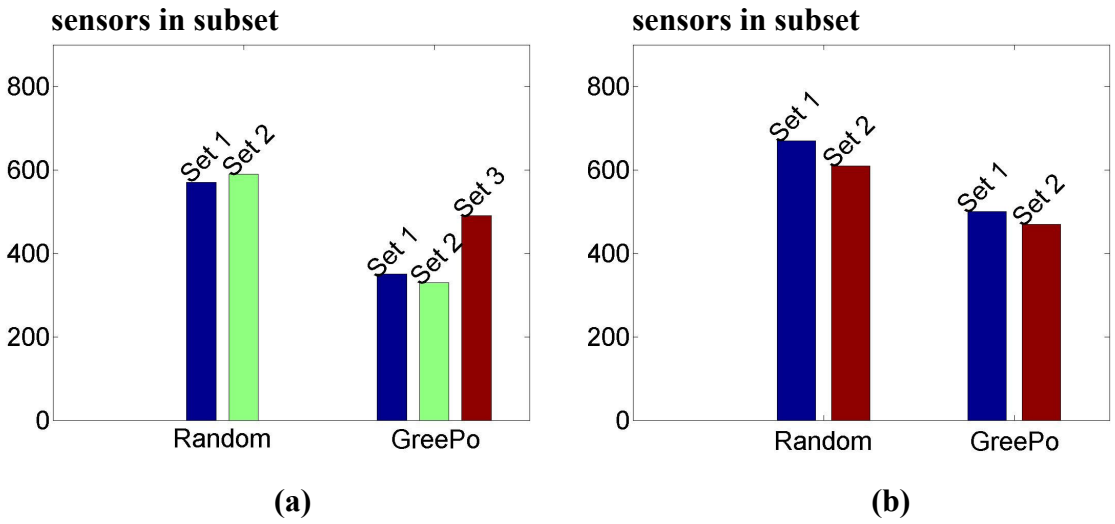

Figure 3.20: Sizes of subsets devised for $N = 1500$ and (a) a stationary process (b) a non-stationary process

Figure 3.20 shows the exact sizes of subsets obtained for $N = 1500$ and each of the two cases of underlying process statistics, for random selection and *GreePo*. Random selection devised 2 subsets for both stationary and non-stationary data, while *GreePo* devised 3 and 2 subsets respectively. The sizes shown are those before assigning unselected sensors among devised subsets, upon termination of the selection algorithms. The significantly larger size of the last subset for *GreePo* in the case of stationary data can be attributed to the characteristics of the uniform deployment used. As certain locations have been 'occupied' by previously constructed subsets, the last subset inevitably contains sensors in swarms around these locations.

In general, efficiency is improved with more available subsets or equivalently with smaller subset sizes. It can be seen that our scheme greatly improves on the efficiency of an unscheduled network (consisting of only one set) by 3 and 2 times for stationary and non-stationary data respectively. Additionally, in the former, it also

improves on the number of subsets devised in comparison to random selection by 50%. The improvement can be explained by observing subset sizes. For example, the best subset devised with *GreePo* consists of 330 sensors. This is a 42% reduction over the 570 sensors comprising the best random subset. For non-stationary data, although there is no additional subset devised, the actual sizes of subsets are smaller: overall 970 sensors are employed by the two subsets of *GreePo* which represents a 24% reduction over the 1280 sensors of random selection.

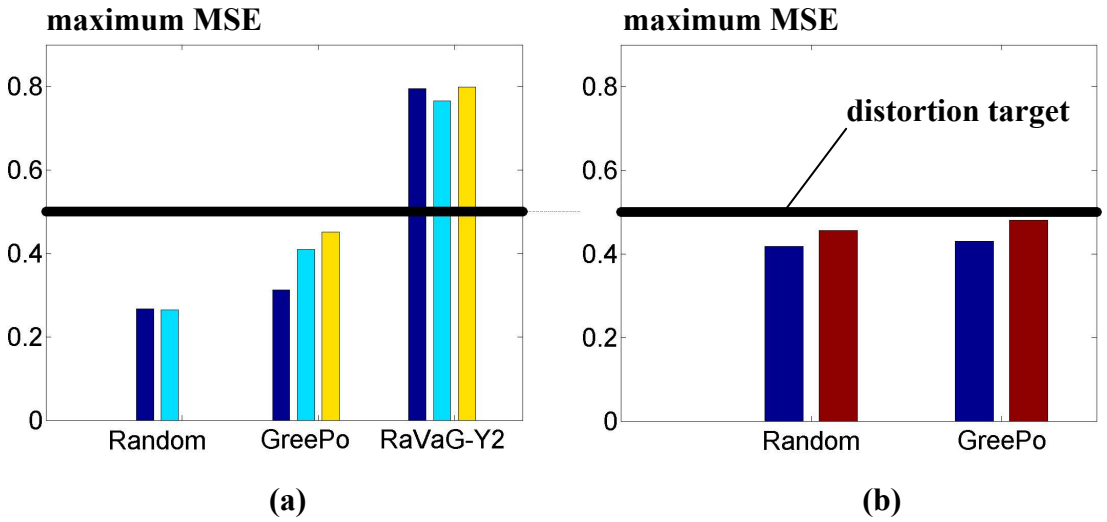

Figure 3.21: Spatially maximum MSE performance, for $N = 1500$ and (a) a stationary process (b) a non-stationary process

Figure 3.21 shows the maximum MSE incurred over space, for subsets obtained with random selection and *GreePo*. Test data consisted of N process realizations generated with the same procedure as the training data. The observation field F was discretized with a test grid of 100x100 locations and distortion was checked on this grid. To mitigate edge effects, locations within a small distance (~0.5 m) from the edge of the observation field were disregarded for distortion calculation. For proof of

concept we also ran the *RaVaG* subset selection algorithm with score function $Y_2(\cdot)$, targeted at spatially averaged distortion. The algorithm devised nine subsets in total (see Figure 3.8), but only the best three have been included in Figure 3.21. It can be seen that subsets devised with both random selection and *GreePo* strictly adhere to the maximum distortion target of 0.5, while those devised with the average distortion algorithm do not. Note also that distortion shown in Figure 3.21 is no longer instantaneous as in Figures 3.9 and 3.13.

Our next step was to study the energy consumption of our proposed algorithm. Data gathering experiments were set up exactly in the same manner as in Section 3.9.1.4, based on an equivalent sensing range of 4.4 m. The overhead incurred for communicating subset membership to individual nodes is identical to that shown in Figure 3.16. Figures 3.22 (a) and (b) depict data gathering packet counts incurred by *GreePo*, random selection and an unscheduled network for both stationary and non-stationary data. The second subset devised by *GreePo* and random selection was employed in each case.

A subtle point here is that, in the non-stationary case, devised subsets were not finalized with redistribution of residual sensors (as has been the rule in all results so far). This was done to preserve the advantage over randomly selected subsets, which would have been obscured with such redistribution, since the total number of available subsets is the same. This is also the reason why energy expended by the *GreePo* subsets appears to be roughly the same for the stationary and non-stationary process case. The final reductions in packet counts for these cases are 89% over an unscheduled network and 42% and 24% respectively over random selection.

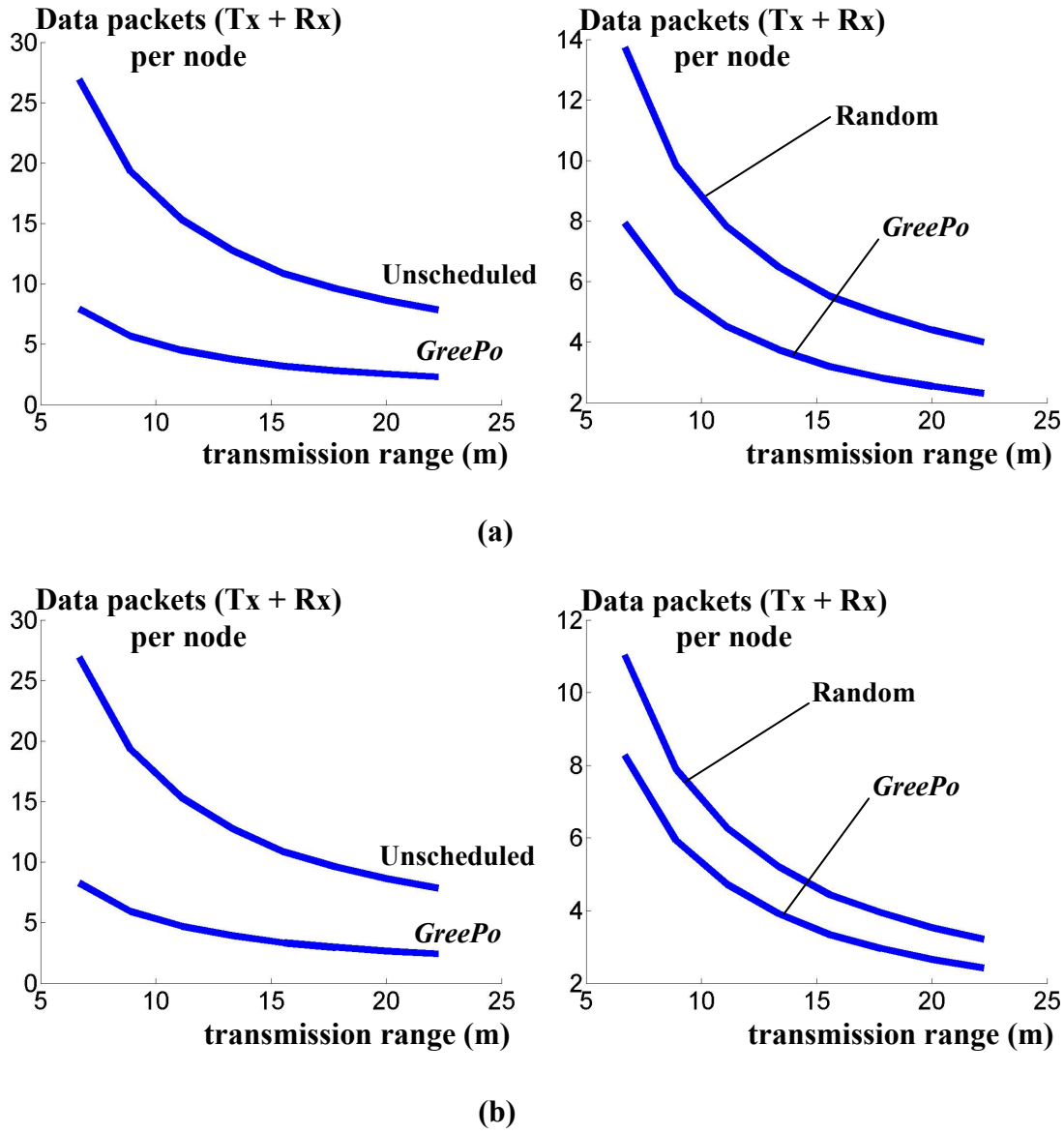

Figure 3.22: Transmissions and receptions of data packets for *GreePo* compared to random selection and an unscheduled network and (a) a stationary process (b) a non-stationary process

We finally applied our *GreePo* algorithm to the temperature dataset from the Sensorscope project [Sen07]. In order to tackle the constraints imposed by these data we had to lower the spatially maximum distortion target from 2 ^{0}F to 5 ^{0}F. An

additional approximation was to test for maximum distortion only at the locations of sensors not in the current subset, i.e., as opposed to a regular grid of locations. This was done without loss of generality, since real data were available only at the locations of the sensors anyway and any interpolation scheme would have to rely on them. With this setup, random selection devised two subsets consisting of 44 and 39 sensors and *GreePo* three subsets consisting of 40, 15 and 28 sensors. The maximum MSE performance of these subsets is shown in Figure 3.23.

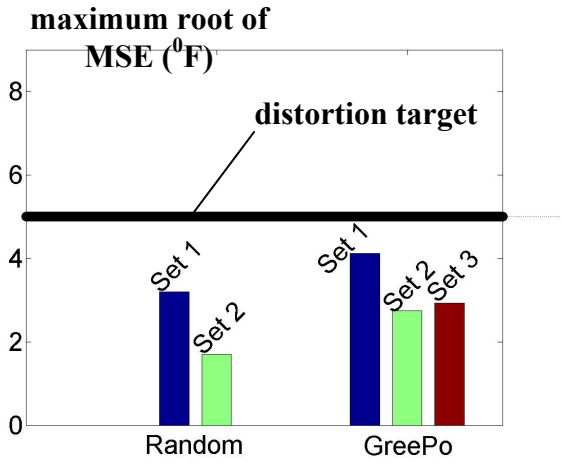

Figure 3.23: Spatially maximum MSE performance for real temperature data

Overall, it can be concluded that *GreePo* achieves the desired distortion performance either with a larger number of subsets or with subsets of smaller size, compared to simpler selection schemes. It is therefore a strong choice when min-max interpolation objectives are involved.

3.10 Related Work

We now provide a brief overview of related work. Such work can be roughly categorized into two major themes: work within the sensor networks community related to the specific problem of sensing topology management itself and work within the signal processing community related to the specific approaches proposed in this chapter to tackle it.

3.10.1 Sensing Topology Management

The idea of sensing topology management, i.e., keeping fewer sensors active at each time instant in order to accomplish a task, is directly related to the redundancy elimination aspect of our work. So far, efforts have mainly focused on event detection applications. The problem of whether or not each point in the observation field is sensed by at least k sensors has been examined in [Hua03]. Many authors have proposed resilient protocols that maintain k-coverage for any point on the field at all times, either in a centralized [Sli01] or distributed [Ai07], [Car04] manner. Set k-cover algorithms [Abr04], [Sli01] aim at obtaining subsets of sensors, such that each is individually capable of k-coverage. These have been the main inspiration for our own work. The problem of simultaneous coverage and wireless connectivity has been investigated in papers such as [Hua03], [Zha05]. Attention has also been given to error resilient coverage [You07]. All of these approaches define coverage on the basis of a circular sensing range. The main drawback is that sensing range has no direct physical interpretation in a sampling-interpolation setting, as opposed to an event detection

setting. In addition, it is not clear how the sensing range can be defined online in a newly deployed network at unfamiliar terrain.

There exists work in this area that forfeits the sensing range assumption. Researchers have considered a Gaussian point source, which could represent time progression of an event, and the correlated random field created around the point. Some have used this type of formulation to define correlation regions of roughly equally informative sensors and partition the network based on a Voronoi tessellation, so that it can best estimate the value of the point source [Vur06]. In a similar vein, other authors have looked at the problem of detecting the source, i.e., whether an event exists or not. Treating detection as a statistical hypothesis problem, they have proposed optimal placements and scheduling of sensors [Sun05]. However, although cognizant of an underlying random process, this work still only targets an event driven type of application, which is different from constructing an entire spatial map. Similarly, the orthogonalization approach of [Qua07] addresses efficient estimation of a deterministic vector parameter with a sensor network, as opposed to prediction of a physical quantity over the entire field.

Redundancy in sampling type applications has been considered, but from a point of view significantly different from ours. Koushanfar et al. [Kou06] aim at pairing sensors, such that the readings from one can be used to predict those of the other. However, they do not study the effect on the overall interpolation quality, but only consider predicting the value at a specific sensor location.

3.10.2 Geostatistics

A relevant theme in statistics and especially geostatistics literature is sampling design: finding the best locations to sample, out of a finite set of possible locations [Kun03]. This is often coupled with Bayesian models for the observed data [Dig06]. An application that has been examined in the context of sensor networks is that of optimum sensor placement. Guestrin et al. have developed algorithms within a Gaussian process framework [Gue05], and have extended their approach to also take communication cost among proposed locations into account [Kra06]. The main idea is to gather data from an initial expert placement of sensors, estimate continuous mean and covariance functions for the Gaussian process, and then redeploy sensors at optimal locations based on the concept of *mutual information*. A crucial difference between this method and our topology management is that, in our case, sensor locations cannot be changed, i.e., they are restricted to initial ones. More recently, a method for selection of monitoring stations based on Bayesian entropy was proposed [Fue07]. The method does not rely on redeployment of stations, but rather estimates a parameterized covariance function and associated entropy using data from those stations already deployed. However, in our case multiple disjoint subsets of sensors need to be devised as opposed to a single one. An additional drawback of both aforementioned methods is that estimation of the continuous covariance function [Gue05] or of the posterior distribution of its parameters [Fue07] is a much more intricate procedure than simple manipulation of Hilbert space inner products as done here. Furthermore, the Bayesian method [Fue07] still relies on entropy to select stations, a criterion which can often lead to unbalanced selections [Gue05].

It is worth mentioning however, that the mathematical form of the $Y_1(\cdot)$ and $Y_3(\cdot)$ score functions for our *RaVaG* algorithm of Section 3.6, are identical to that for greedy maximization of entropy and mutual information respectively [Gue05], if the random variables in the primary subspace are assumed jointly Gaussian. The connection between the methods would be interesting to investigate further, but goes beyond the scope of this dissertation. The theme of network design for interpolation will be revisited in Chapter 4, with a more thorough account of related methodologies.

3.10.3 Compressive Sensing

A fairly recent development in the sensor networks regime is *compressive sensing*. Consider an n-dimensional data vector with elements in $\Re$ which is compressible in some basis of $\Re^n$. This means that it can be recovered with small error from its projections, i.e., its Euclidean inner products, with k out of the n vectors comprising the basis. The basic premise of compressive sensing is that, in such a case, the data vector can also be recovered from its Euclidean inner products with *k random* n-dimensional vectors, i.e., not necessarily with the specific vectors forming the basis. With this technique, each sensor can effectively multiply its datum with a random value (in case of a scalar datum) or with a random vector (in case of vector data), with all such vectors being generated locally at each sensor with the same random seed. Sensors then only have to forward these projection values to the data center instead of their complete data.

Redundancy in signals measured by the sensors has been considered in [Dua06], where all sensors report compressed data. However, authors thereof do not specifically

aim at providing distortion guarantees for spatial interpolation as we do here, but rather aim at reconstruction of individual sensor signals in time. As such, they focus more on temporal redundancy whereas we focus on spatial redundancy. Additionally, compressive sensing results in compressed data for all sensors in the network, i.e., all of them have to send data packets, albeit smaller. Our approaches explicitly leverage spatial redundancy and reduce the number of data packets produced by the network, as opposed to data packet sizes, to meet the demands of a specific application.

The spatial dimension in sensor readings has been exploited in two other compressive sensing papers namely [Baj06], [Wan07]. In both of these, the collection of n real data values from all sensors at a single time instant forms the data vector. In the former paper, the goal is to communicate this data vector to the data center: using an uncoded coherent transmission scheme, sensors report their projection values during k channel uses, where k is the number of necessary compression coefficients. The total number of necessary channel uses is hence reduced from n to k. However, since the random projection vectors are not sparse, at each channel use all n sensors still have to transmit data, which represents a fundamental difference with our objective of reducing the total number of data packets. In the latter paper, the goal is to distribute random projection values throughout the network in such a way as to be able to reconstruct the data vector by querying any k sensors. The distributed protocol used, incurs a normalized (i.e. per sensor) packet overhead that is logarithmic in the number of sensors in the network, but is also incurred every time the values measured by the sensors change. The associated energy cost is combined with the cost of actually querying k sensors, leading to a potentially large energy overhead, if the measured

values change rapidly. Thus the major difference of this scheme with what we have proposed so far is that it primarily targets robust data storage rather than real time monitoring and interpolation.

Compressive sensing overall provides a fruitful ground for interaction with our schemes *after* disjoint sensor subsets have been devised, because it can further compress the data of chosen sensors. However, exploring this is outside the scope of this dissertation.

3.10.4 Sparse Methods

A tremendous amount of effort has been devoted in efficiently obtaining sparse approximations of data sets. A topic relevant to our work is Sparse Principal Components Analysis [Asp04], [Mog06], [Zou06]: the eigen-components (i.e. dimensions) of a data matrix are sought as linear combinations of only a subset of all initial dimensions (i.e. columns). The decomposition is such that derived components capture maximum variance in the data. In our particular problem setup, Sparse Principal Components Analysis is not applicable, since it is based on the criterion of capturing variance. In other words, sensors that measure highly correlated values are all likely to be selected, as long as their values are characterized by high enough variance. Intuitively, our goal of devising equivalent monitoring subsets relies on the principle that highly correlated sensors should at least be in different subsets.

Another related line of research is subset selection for linear regression in statistics. The goal of linear regression is to approximate a dependent or *response* random variable by a linear combination of *explanatory* random variables. Subset

selection aims at keeping only a subset of representative variables to perform approximation. Practical greedy methods such as forward selection and backward elimination are detailed in [Mil02]. Recent work has investigated approaches to the problem when multiple response variables are entailed [Sim06] as well as for cases where the statistical correlation among explanatory variables adheres to certain structure [Das08a]. Subset selection for linear regression is also not applicable to our particular problem since there are no fixed response variables. Essentially, all sensors are response variables. Our goal is to approximate the space spanned by all response variables with the space spanned by an appropriate subset of these.

On the other hand, the signal processing community has focused on a more concrete systems problem: sparse approximation of a finite-dimensional real or complex vector with a *dictionary* of vectors, as described in the Section 3.4. The basic problem can be formulated as [Wip04]:

$$\boldsymbol{t} = \Phi \cdot \boldsymbol{w} + \boldsymbol{\varepsilon} \tag{3.27}$$

where $\boldsymbol{t} \in \Re^{K}$ is a vector of targets, $\Phi \in \Re^{K \times M}$ with $M \geq K$ is a dictionary of M columns or atoms that have been observed or determined by experimental design, $\boldsymbol{w}$ is a vector of unknown weights and ε is noise. After observing $\boldsymbol{t}$ and given Φ, the goal is to estimate $\boldsymbol{w}$, taking into account the prior belief that $\boldsymbol{t}$ has been generated by a sparse subset of the atoms, i.e., that most of the elements in $\boldsymbol{w}$ are equal to zero. The general problem has been proven to be NP-hard, as mentioned in Section 3.4 [Dav97]. Relevant approaches include the forward greedy matching pursuit and the well known Orthogonal Matching Pursuit [Dav97], convex minimization of the ℓ_1 norm of the

vector of coefficients $\boldsymbol{w}$ (sometimes called Basis Pursuit) [Don03], iterative minimization of general ℓ_p norms of $\boldsymbol{w}$, with $p \leq 1$ (known as the FOCUSS algorithms) [Rao03], as well as backward elimination [Cot02]. An effective approach based on the use of data-adaptive priors for the weights, appears in [Wip04], and is known as Sparse Bayesian Learning.

The case more relevant to our setup is that of multiple response vectors:

$$T = \Phi \cdot W + \boldsymbol{\varepsilon} \tag{3.28}$$

where T is now an aggregate of L target columns and W is the aggregate of corresponding weights, with the added constraint that W has a minimal number of non-zero rows. This essentially means that the responses T are all supposed to have been generated by the same dictionary atoms, perhaps with different coefficients. The single vector methods have been extended to address this problem known as *simultaneous sparse approximation*: a few representative examples are Simultaneous Orthogonal Matching Pursuit (SOMP) [Tro06], minimization of diversity measures for the matrix of weights W [Cot05], as well as the extension of Sparse Bayesian Learning to the multiple response case [Wip07].

Simultaneous sparse approximation algorithms thus devise representations for a deterministic signal, usually a Euclidean matrix with real or complex elements, based on an external dictionary. In our problem setup, there is no straightforward notion of a deterministic matrix to approximate, since elements of the involved Hilbert space are abstract quantities (random variables). Applicability of such algorithms can be sought if one tries to sparsely approximate the data matrix B_0, i.e., the matrix of

measurements from all sensors gathered during the learning phase, also using B_0 as the dictionary. This inherently suffers from a generalization problem: existing methods would try to sparsely approximate the data matrix B_0 itself, instead of trying to minimize the error induced by selected subsets on *future data*. By contrast, our own *RaVaG* algorithm uses the data matrix B_0 only for estimation of inner products in the Hilbert space of second order random variables and has better generalization performance as shown in Section 3.9.

A major additional problem is that existing sparse approximation methods eventually produce one as opposed to many approximating subsets for the signal. Recall that the data matrix B_0 of eq. (3.8) effectively constitutes both the dictionary and the target vectors. The target vectors are hence not to be approximated by an external dictionary, but by 'subsets of themselves'. An equivalent interpretation is that the dictionary is to be approximated with itself. In this case, it is not clear how to use algorithms that minimize the ℓ_p norm of the weight vector, after the first subset has been constructed: for the usual case $\Theta = K = N$ for example, some columns of the dictionary will have to be excluded from the selection process, rendering $K > M$ in eq. (3.28), and the problem automatically ceases to adhere to the definition of sparse approximation. Simple greedy methods such as Orthogonal Matching Pursuit on the other hand, are more amenable to modification so as to devise multiple subsets. This modification has led to the *MaG* algorithm presented in Section 3.7.

3.10.5 Min-Max Objectives

Min-max problems have not yet received as much attention in the literature as problems involving averages. A possible reason for this is that average objectives (e.g. eq. (3.22)), unlike maximum ones (e.g eq. (3.23)), are convex and, as such, susceptible to a rich pool of convex optimization methods.

A state-of-the-art geostatistical min-max design approach appears in [Wie05]. However, the goal thereof is to minimize maximum distortion not over spatial locations as we do here, but over the worst case misspecification of the covariance model. Authors in [Das08b] examine worst case prediction of the value of aggregate functions against hard constraints as quantified by a distance metric. This is different from our specific definition of distortion in eq. (3.23) and our goal to optimize over spatial locations. In [Kra07a] authors select observation locations against min-max objectives. Spatial locations are taken into account, albeit through a collection of deterministic models (specifically, a collection of submodular functions) one for each candidate location. These may be difficult to estimate in practice or may be altogether unsuitable in certain interpolation scenarios (e.g. submodularity may not always hold). On the other hand, we have presented in Section 3.8 a method that directly takes into account the stochastic interdependencies of sensor values in a way specifically geared towards interpolation.

3.11 Summary

In this chapter we have presented strategies towards efficient operation of an a-priori deployed wireless sensor network for the task of spatial interpolation. The

problem is *organizational* in nature: the sensor network is initially overdeployed, i.e., some sensors are redundant and disjoint subsets must be devised such that each of them is individually capable of achieving the desired interpolation accuracy. Sensing functionality is rotated among these subsets, forming the notion of sensing topology management. Our main motivation has been to reduce the amount of raw data packets produced by the network at any given time.

We have proposed a number of schemes which addressed two different interpolation objectives, namely spatially averaged MSE and spatially maximum MSE. Our approaches are generic enough to effectively handle spatial non-stationarities in the observed physical processes. Experiments on real as well as synthetic data have shown substantial gains in the number of devised subsets that can support a user specified target distortion compared to simpler sensor selection techniques. Specifically regarding spatially averaged distortion, the *RaVaG* algorithm achieved the best overall performance. This translates to a significant reduction in the number of data producing sensors at any point in time. Associated energy consumption has indeed been shown to be significantly improved compared to that of an unscheduled network. By using subsets in sequential activation, the network can therefore remain operational for a longer duration of time.

3.12 Acknowledgements

This chapter is, in part, a reprint of material published in three articles: “A Distortion-Aware Scheduling Approach for Wireless Sensor Networks”, appearing in the Proceedings of the Second IEEE International Conference on Distributed

Computing in Sensor Systems, DCOSS 2006, Lecture Notes in Computer Science 4026, Springer 2006, ISBN 3-540-35227-9; "Leveraging Redundancy in Sampling-Interpolation Applications for Sensor Networks" appearing in the Proceedings of the Third IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2007, Lecture Notes in Computer Science 4549, Springer 2007, ISBN 978-3-540-73089-7; and "Energy Efficient Min-Max Spatial Monitoring with Wireless Sensor Networks" appearing in the Proceedings of the IEEE Wireless Communications and Networking Conference, 2009. It is also, in part, submitted for publication in the Springer Journal of Signal Processing Systems under the title "Efficient Sensing Topology Management for Spatial Monitoring with Sensor Networks" and in ACM Transactions on Sensor Networks, under the title "Leveraging Redundancy in Sampling-Interpolation Applications for Sensor Networks: A Spectral Approach". In all cases, I was the primary researcher and author and Curt Schurgers supervised the research.

CHAPTER 4

4. SPATIAL INTERPOLATION: THE PROACTIVE CASE

4.1 Background

In this chapter, we will examine approaches to the proactive management scenario of Section 1.6. Unlike the reactive methodology of the previous chapter, the scenarios of interest here are ones where there is no network already deployed. The fundamental difference is that, instead of managing redundant sensors, here we are required to guide the deployment from scratch, i.e., to design how exactly sensors are going to be deployed to support the specific purpose of spatial interpolation. Whereas in the reactive management scenario the goal was to minimize the amount of data packets injected in the network, the important goal here is to minimize the total number of sensors deployed to begin with, because physically deploying them induces material as well as programming cost. Therefore the problem can be succinctly expressed as *minimizing how many sensors to place and decide where to place them, so as to achieve an accurate enough interpolation over time*.

In developing solutions for this problem, our main assumption is that we possess no information on the statistics of the phenomenon to be monitored, prior to deployment. This is the case in many real-life situations and was implied when reactively managing the network in Chapter 3 as well. However, the crucial difference in the present scenario is that there is no existing network to draw data values from

and promptly infer the necessary covariance information. Instead, we have to estimate the statistics by specifically deploying sensors for this purpose. Placements should be done in an intelligent manner, so as to have only a minimum number of sensors deployed in the end.

To achieve this, we propose an *incremental deployment* framework, consisting of two steps:

1. The first step is an *exploration phase*. The goal of this phase is to sequentially deploy sensors at certain locations so as to learn relevant statistics of the physical quantity of interest.
2. The second step is a *design phase*. The goal of this phase is to utilize statistics learned during the exploration phase and sequentially deploy sensors to monitor the physical phenomenon. Sensing at the deployment locations ought to collectively satisfy the quality criteria of the spatial interpolation application.

There are indeed practical cases where such incremental deployments are plausible as well as possible to implement. Consider the example of a wireless sensor network (WSN) monitoring temperature or air flow within a manufacturing plant. Cases such as this one arise frequently in contemporary indoor scenarios where the very architecture or arrangement of objects provides a natural framework for the placements of sensors. An additional factor to take into account is that the size of such networks, at least with current technology, is not large. Straightforward approaches such as placing ‘a sensor in every room’ are both common and realizable. In the

following sections we will further elaborate on our methodology and show how it leads to deployments featuring a minimum number of sensors.

4.2 Objectives

The objective of our methodology is ultimately to choose physical locations such that, if sensors are placed at them, the requirements of the sampling-interpolation application are met. It is often appropriate to consider a discrete set of possible locations to place sensors at, instead of the continuous 2-D physical space. In an indoor temperature monitoring scenario for example, a continuum of possible locations would not be allowed on account of walls or other obstructions.

The setup for the problem follows along the lines of the generic setup for interpolation described in detail in Section 2.2, with one important difference: Here there are N possible locations to deploy sensors at, instead of N sensors already deployed. Locations $\{\boldsymbol{X}_0^p\}$, p = 1…N are distributed over the observation field F and their number N is possibly very large. The two distinct phases of the methodology can then be formalized as follows:

- *Exploration phase*: The goal of this phase is to select a subset $\boldsymbol{X}_e$ of *exploratory locations*, where $\boldsymbol{X}_e \subset \boldsymbol{X}_0$.
- *Design phase*: The goal of this phase is to select a subset $\boldsymbol{X}_m$ of *monitoring locations*, where $\boldsymbol{X}_m \subset \boldsymbol{X}_0$.

A *network design* or simply design is the subset of monitoring locations $\boldsymbol{X}_m$ (as opposed to the exploratory subset $\boldsymbol{X}_e$). A network design will in general be represented

by the Boolean vector $\boldsymbol{m}$ of length N, where each element is 1 for a position bearing a sensor and 0 otherwise. The exact positions comprising the design will be denoted by the tuple $\boldsymbol{X}_{\boldsymbol{m}}^{p} = (x^p, y^p)$, p=1…$|\boldsymbol{m}|$, where $|\cdot|$ is the size of a set. Essentially the monitoring subset $\boldsymbol{m}$ is a subset $\boldsymbol{m}_k$ in the notation introduced for the generic interpolation case at Chapter 2. In what follows we will rely on this notational correspondence, obtained if the subscript $\boldsymbol{m}$ is substituted by the subscript k.

A 'good' network design should be based on how sensor data are ultimately used by the user application. Consider the situation where sensors have actually been deployed at the monitoring locations $\boldsymbol{X}_{\boldsymbol{m}}$. Then an interpolator that is linear on the measured data will have the form (see eq. (2.1))

$$\hat{S}_{\boldsymbol{m}}(\boldsymbol{x},t_i) = \lambda_0(\boldsymbol{x},\boldsymbol{m}) + \sum_{p=1}^{|\boldsymbol{m}|} \lambda_p(\boldsymbol{x},\boldsymbol{m}) \cdot S(\boldsymbol{X}_{\boldsymbol{m}}^{p}, t_i) \tag{4.1}$$

The design problem for spatial interpolation is thus to find a minimum monitoring subset $\boldsymbol{X}_{\boldsymbol{m}}$ of all possible locations $\boldsymbol{X}_0$, so that if sensors are deployed at these locations, the interpolation (4.1) meets a specific distortion criterion. We focus on the following two distortion criteria (see eq. (3.22)-(3.23)):

$$\underset{\boldsymbol{x}\in A}{avg}(D_{\boldsymbol{m}}) \approx \frac{1}{|A|}\sum_{\boldsymbol{x}\in A} E[(\hat{S}_{\boldsymbol{m}}(\boldsymbol{x},t) - \hat{S}_0(\boldsymbol{x},t))^2] \le D \tag{4.2}$$

$$\max_{\boldsymbol{x}\in A}(D_{\boldsymbol{m}}) \approx \max_{\boldsymbol{x}\in A}(E[(\hat{S}_{\boldsymbol{m}}(\boldsymbol{x},t_i) - \hat{S}_0(\boldsymbol{x},t_i))^2]) \le D \tag{4.3}$$

The reasoning behind this choice of criteria as well their interpretation have been extensively discussed in Sections 3.2, 3.3 and 3.8.

The minimum Mean Squared Error (MSE) inflicted by the best linear interpolator of the form (4.1) is given by (see eq. (2.12)):

$$MMSE(\boldsymbol{m},\boldsymbol{x}) = \min_{\hat{S}}\{E[(\hat{S}_{\boldsymbol{m}}(\boldsymbol{x},t_i) - S(\boldsymbol{x},t_i))^2]\} = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{c}_{\boldsymbol{m}}^T(\boldsymbol{x}) \cdot G_{\boldsymbol{m}}^{-1} \cdot \boldsymbol{c}_{\boldsymbol{m}}(\boldsymbol{x}) \tag{4.4}$$

where $G_{\boldsymbol{m}}$ and $c_{\boldsymbol{m}}(\mathbf{x})$ have been effectively defined in eq. (2.11). The important conclusion that can be drawn from the form of these equations is that in order to apply algebraic operations, such as orthogonal projection, to our design setting, it is sufficient to know the spatial covariance function $C(\cdot, \cdot)$.

4.3 Exploration Phase

4.3.1 Fundamentals

As mentioned previously, the stepping stone to utilizing vector operations in our Hilbert space framework is knowledge of the spatial covariance function $C(\cdot, \cdot)$. In most real life situations the covariance function will be not known a-priori. Our reactive management scenario dealt with this hurdle by initially overdeploying and then using data from all available sensors to compute necessary covariance values. Here nevertheless, there is no pre-existing network to rely on; thus we must first utilize an exploration phase to obtain an estimate of the covariance function and subsequently use it to design. As a sidenote, in practical cases where the covariance function is reliably known before deployment, e.g. from legacy sensory mechanisms, the exploration phase can be skipped altogether.

In the most general case, the spatial covariance is a $\Re^4 \rightarrow \Re$ function, i.e., it is mapping every pair of positions in 2-D space to a real value. An important special case however, arises when the physical phenomenon can be regarded as isotropic. The

isotropic model has been widely used to describe geostatistical and atmospheric data [Ber01], [Ste99]. For this type of physical phenomenon, the spatial covariance between any two locations depends only on their distance, i.e., $C(\boldsymbol{x}_1, \boldsymbol{x}_2) = C(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|)$ and $\|\cdot\|$ denotes Euclidean distance. In the present section we focus on this special case. It should be stressed that the general mathematical tools used here are applicable to a higher number of dimensions. The details thereon can be considered as extensions to our present work.

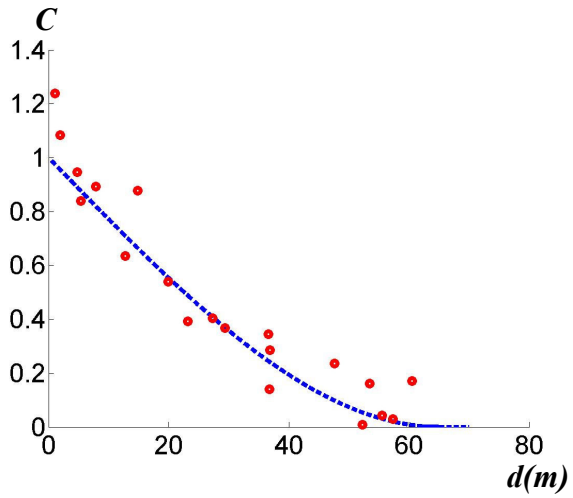

Figure 4.1: Ground truth covariance curve (dashed line) and estimated covariance points from exploration (dots).

The isotropic covariance function is essentially a curve *C(d)* that describes the behavior of spatial correlation with distance *d*. The objective of our exploration procedure is to obtain covariance values at known distances and then deduce a reliable estimate of the ground truth curve *C(·)* by interpolating these values. An example is shown in Figure 4.1. Estimating *C(·)* essentially *constitutes a new interpolation problem* albeit at a higher conceptual level than interpolating measurements: we are

now trying to devise a reliable model to describe the *correlation* between measurements.

In the remainder of this section, we elaborate on how data points are chosen and interpolated to lead to the estimate of *C(d)*. Exploration starts with two sensors deployed at random in the observation field, at locations $\boldsymbol{X}_e^1$ and $\boldsymbol{X}_e^2$. These sensors simultaneously sample the physical phenomenon for a time period of Θ time instants. The time series of values from each sensor are stacked together in a Θx2 matrix B_2. We call this the *exploration data matrix* based on two sensors:

$$B_2 = \begin{bmatrix} S(\boldsymbol{X}_e^1, t_1) & S(\boldsymbol{X}_e^2, t_1) \\ \dots & \dots \\ S(\boldsymbol{X}_e^1, t_\Theta) & S(\boldsymbol{X}_e^2, t_\Theta) \end{bmatrix} \tag{4.5}$$

The data matrix can be thought of as a finite dimensional approximation to the infinite dimensional random variables corresponding to the locations of the sensors. Similarly, the *exploration covariance matrix* $\hat{G}_2$ is an approximation to the Grammian matrix of actual inner products between these two sensors:

$$\hat{G}_2 = \frac{1}{\Theta - 1} \cdot (B_2 - \hat{\mu}_2)^T \cdot (B_2 - \hat{\mu}_2) \tag{4.6}$$

where $\hat{\mu}_2$ is a matrix with all rows equal to the row vector of means of the columns of B_2, i.e., the empirical means of the random variables corresponding to the sensors. After computing the exploration covariance matrix, and since the locations bearing sensors are known, we essentially possess data values *y* for two different distances *d*, i.e., $d = 0$ and $d = d_{12}$, where d_{12} is the distance between the two locations. Note that one of these distances, $d = 0$, typically bears as many different values as the number of

deployed sensors (in this case two), each corresponding to variance at the respective location.

To explore, we sequentially add sensors to this initial set of two. There are two possibilities for deployment of additional sensors: either we can have control over the deployment locations or not. Relevant scenarios mentioned in Section 4.1 mainly involve settings in limited areas (e.g. indoors), where possessing such control is actually possible. The very sequential nature of our methodology also implies that some control over actual deployments would be beneficial from a logistics point of view (i.e. physically going into the observation field and throwing out a sensor at random every time seems wasteful). Hence we will first focus on the case where we can indeed choose exactly where to add a new sensor and defer the uncontrollable case for later on.

After each addition, all sensors simultaneously sample the physical phenomenon for Θ time instants. With k sensors in the exploration subset, the exploration data matrix B_k has size Θxk and provides $n = k\cdot(k - 1) / 2 + k$ data values y at $k\cdot(k - 1) / 2 + 1$ different distances. We will denote the pairwise Euclidean distances between any two sensors in the exploration subset as d_{lq}:

$$d_{lq} = \left\| \boldsymbol{X}_{\boldsymbol{e}}^{l} - \boldsymbol{X}_{\boldsymbol{e}}^{q} \right\| \tag{4.6}$$

In what follows, all available data values (including sensor variances) will be denoted by $\{y_j\}_{j=1\ldots n}$. The problem posed by exploration is then how to select as few locations as possible to deploy sensors at, so as to obtain a good enough estimate of the covariance curve $C(d)$ based on the values $\{y_j\}$. This is essentially a new interpolation

problem, with the important characteristic that statistics of the data values are not available. Therefore, straightforward methods, such as statistical least squares, cannot be applied.

4.3.2 Greedy Kernel Regression

Our way to model the actual covariance curve with respect to the data values $\{y_j\}$ is kernel regression [Bow97]. The key element of kernel regression is to describe the data points y by a curve representing the ground truth, contaminated by a white noise term with mean 0 and variance σ^2:

$$y_j = C(d_j) + \varepsilon_j \tag{4.7}$$

Obtaining an estimate $\hat{C}(d)$ of the ground truth curve, is the main objective of kernel regression. For reasons that will be elaborated shortly, our specific choice in this family of estimators is local linear kernel regression. Extensive details thereof can be found in [Bow97]. Here, we only describe those aspects essential to our scheme.

Local linear regression produces a curve which, for each argument d, has the value of a straight line fitting especially those data y whose support is close to d. This proximity is quantified by a positive-definite kernel $w(\cdot; h)$, where h is a bandwidth parameter. In what follows we will use as kernel the Gaussian density function with mean 0 and standard deviation h. Specifically, with n available data values, the following least squares problem must be solved for each d:

$$\hat{C}(d) = \arg\min_{a,b} \sum_{j=1}^{n} \{y(d_j) - a - b \cdot (d_j - d)\}^2 \cdot w(d_j - d; h)) \tag{4.8}$$

This ultimately yields an estimate that is linear in the data:

$$\hat{C}(d) = \sum_{j=1}^{n} u_j \cdot y_j \tag{4.9}$$

The reliability of this estimate for a set of points $\{y_j\}$ at distances $\{d_j\}$ can be expressed through the MSE with respect to the probability distribution of the noise:

$$E[(\hat{C}(d) - C(d))^2] = (\sum_{j=1}^{n} u_j \cdot C(d_j) - C(d))^2 + \sigma^2 \cdot \sum_{j=1}^{n} u_j^2 \tag{4.10}$$

The first term in the above sum is the squared bias while the second is the variance of the estimate. Equation (4.10) cannot be computed for an arbitrary choice of d_j without knowledge of the actual covariance curve *C(d)* [Bow97], [Har04]. Only asymptotic expressions have been derived in related literature. A reliability measure that is independent of the particular choice of $\{d_j\}$, is the Mean Integrated Square Error (MISE) [Bow97]:

$$MISE(h) = \int E[(\hat{C}(d) - C(d))^2] \cdot f(x) \cdot dx \tag{4.11}$$

where *f(x)* denotes the density of arguments $\{d_j\}$. This will be used as a stopping criterion in our exploration, in a manner that will soon be elaborated.

The exploration problem now becomes that of selecting as few locations as possible out of *N* to make the estimate $\hat{C}(d)$ as reliable as possible, according to eq. (4.11). This is in general a hard combinatorial problem: optimal solution requires enumeration of all possible subsets and evaluation of the reliability of the estimate resulting from each subset. The number of possible subsets is exponential in *N*, which can be in the order of thousands.

Here we propose a greedy exploration scheme instead. Our scheme relies on two important properties of local linear regression. Firstly, asymptotically, the bias term in eq. (4.10) does not depend on the arguments $\{d_j\}$, but only on the bandwidth h. This means that, asymptotically, the flexibility of the kernel regression estimate (which controls the bias) can only be enhanced through the bandwidth parameter and is insensitive to the addition of more data points. This is the *design adaptive* property of local linear regression [Bow97]. However, for small constant h, addition of more data points does affect the variance, i.e., the reliability of the estimate. Secondly that, asymptotically, the variance of local linear regression is the same as the variance of local constant regression, i.e., the regression resulting from the optimization (4.8) by setting $b = 0$. Interestingly, in the local constant case, this variance does not depend on the measured data directly. Instead, it depends on the arguments $\{d_j\}$ and the bandwidth parameter h and can be analytically expressed in the following form:

$$V_k(d) = \sigma^2 \cdot \sum_{j=1}^{n} u_j^2 = \sigma^2 \cdot \frac{\sum_{j=1}^{n} w^2(d - d_j; h_{k-1})}{(\sum_{j=1}^{n} w(d - d_j; h_{k-1}))^2} \tag{4.12}$$

The subscript k is indicative of the fact that the variance $V_k(d)$ is computed with k exploratory sensors (recall from the previous subsection that $n = k \cdot (k - 1) / 2 + k$). Note also that in the above expression we have allowed the bandwidth parameter to change with each new sensor addition.

To minimize the variance one would equivalently have to maximize the following fitness function $Q_k(d)$:

$$Q_k(d) = \frac{(\sum_{j=1}^{n} w(d - d_j; h_{k-1}))^2}{\sum_{j=1}^{n} w^2(d - d_j; h_{k-1})} \tag{4.13}$$

Intuitively, maximizing this fitness function places more kernels, i.e., more data points close to those distances *d* that have the largest variance. An important observation is that the number of data points *n* is quadratic in the number of sensing locations *k*. Hence, their number grows fast and, although the above formulation depends on asymptotics, eq. (4.13) still gives pragmatic results even for a small number of sensing locations. This will be shown in Section 4.6.

Our entire exploration scheme *GreeX*, is shown in Figure 4.2. It first finds the *d* for which $Q_k(d)$ is minimum (see eq. (4.13)) within the range $[2 \cdot h, \max_j \{d_j\} - 2 \cdot h]$ (to avoid edge effects) and then selects as next monitoring location the one that maximally increases this $Q_k(d)$. Optimization is essentially performed over the variance part of the MSE of eq. (4.10), which depends on the actual data arguments $\{d_j\}$. Exploration stops when addition of a new location does not significantly change the MISE (Mean Integrated Square Error) of eq. (4.11). The MISE is used to indicate convergence because it asymptotically does not depend on $\{d_j\}$, and hence the same convergence threshold can be used to characterize and compare various exploration strategies. MISE can be approximately computed through the cross-validation (or 'leave-one-out') formula [Bow97]:

$$MISE(h) \approx CV(h) = \frac{1}{n} \cdot \sum_{j=1}^{n} (y_j - \hat{C}_{-j}(d_j)) \tag{4.14}$$

where $\hat{C}_{-j}(d_j)$ is the estimate (4.9) computed at d_j when the data point y_j has been excluded. By virtue of the Law of Large Numbers, this expression is guaranteed to converge to the true MISE, within a constant [Bow97].

An approximately optimal bandwidth parameter h can also be computed from the cross-validation formula every time a new sensing location is added. This is done in Line 16 by trying several different values of h and selecting the one that minimizes eq. (4.14). It can be shown that the bandwidth thus found is asymptotically decreasing with the number of available data points and therefore allows for a small bias in the estimate [Bow97].

1 **Input**: $\boldsymbol{X}_0$, δ
2 **Output**: subset of locations $\boldsymbol{X}_e$
3 **Initialization**:
4 Select two locations from $\boldsymbol{X}_0$ at random
5 Compute $\hat{C}(d)$ from eq. (10)
6 $h_2 \leftarrow \arg \min_{h \in (0,\ \max\{d_j\})} (CV(h))$
7 Compute $Q_2(d)$ from eq. (15)
8 **begin**
9 k $\leftarrow$ 2
10 **repeat**
11 $d_{min} \leftarrow \arg \min_{d \in [2 \cdot h_k,\ \max_k\{d_k\} - 2 \cdot h_k]} (Q_k(d))$
12 $\boldsymbol{X}^* \leftarrow \arg \max_{\boldsymbol{X}_0} (Q_{k+1}(d_{\min}))$
13 $\boldsymbol{X}_e \leftarrow \boldsymbol{X}_e \cup \boldsymbol{X}^*$
14 $\boldsymbol{X}_0 \leftarrow \boldsymbol{X}_0 \setminus \boldsymbol{X}^*$
15 $k \leftarrow k + 1$
16 $h_k \leftarrow \arg \min_{h \in (0,\ \max\{d_j\})} (CV(h))$
17 **until** ($|CV(h_k) - CV(h_{k-1})| \leq \delta$)
18 **end**

Figure 4.2: Greedy Exploration (*GreeX*) algorithm

As a final remark, if no control over sensor deployment locations can be exercised, we can resort to a completely random exploration scheme:

Random exploration: Select k locations at random to comprise the exploration subset $\boldsymbol{X_e}$*. Keep increasing k until the cross-validation criterion (Line 17 of Figure 4.2) is satisfied.*

In section 4.5 we will demonstrate results both for random exploration and *GreeX*.

4.4 Design Phase

When the covariance function *C(d)* has been estimated, the inner product of the space of sensors may also be computed. Hence, the quality of different locations with respect to interpolation can be readily quantified through the orthogonal projection error of eq. (4.4). The general design problem, i.e., selecting a subset $\boldsymbol{X_m}$ of locations to deploy sensors at, is a hard combinatorial problem, even with a-priori known statistics [Dig06], [Gue05]. Optimal solution requires enumeration of all possible sensor configurations and validation of each of them against the design objectives. This specific problem has been extensively studied in the context of geostatistics and has been proved to be hard for a variety of related objectives, such as kriging variance [Zhu06], [Zim06], entropy [Ko95] and mutual information [Gue05]. However, a well established optimal design theory does not yet exist [Zim06]. Good approximation approaches should be pursued instead.

In the following subsections, we briefly describe two such tractable design methods for spatially averaged and min-max monitoring respectively. The approaches are based on tools developed in Chapter 3 for the reactive management scenario: having already deployed a large sensor network and selecting subsets of sensors adequate for interpolation.

A final point of interest is that proposed methods assume the availability of a discrete set of candidate locations (the set $\boldsymbol{X}_0$ of Section 4.2) at which sensors can be exactly deployed. This is the most relevant assumption for scenarios of interest, as has been discussed previously. In the case where sensor placement cannot be controlled, the design problem is no longer deterministic but statistical: the only insight a practitioner can hope to obtain from a design strategy is estimates for the parameters governing the probability distribution of the envisioned deployment. If uniform Poisson sampling is to be used for example, the goal would be to estimate the intensity of the Poisson distribution necessary to provide accurate enough interpolations. A way to tackle this situation would be empirically, through simulation. We would have to simulate actual deployments for many different values of the involved parameters and compute either spatially averaged distortion (eq. (4.2)) or spatially maximum distortion (eq. (4.3)) by virtue of eq. (4.4). Then we would select appropriate values for these parameters so as to meet the target distortion. The prominent issue in this case is how to accurately simulate different types of deployments, i.e., different types of point processes. This is a well studied subject [Mol03] and hence will not be further examined it here. Instead we focus on the case of having to make deterministic choices about where to place sensors.

4.4.1.1 Spatially Averaged Distortion

In this case, we seek locations such that, if sensors are placed at them, the averaged distortion over space adheres to a bound D (see eq. (4.2)). Our method sequentially selects sensing locations until the distortion criterion of eq. (4.2) by virtue of eq. (4.4) is met. Consider, in general, a situation where we are in the process of creating the monitoring design. At this point, the initial sensor locations $\boldsymbol{X}_0$ can be considered as being partitioned in two subsets: 1) the subset H_A of locations already selected by the design; 2) the subset H_R of locations not yet selected by the design. Our algorithm considers all candidate locations η from those not yet belonging to the design. For each of them, it computes the MMSE of eq. (4.4) as if a sensor on location η was to be approximated by sensors placed at all locations H_A or H_R (always excluding the sensor at η):

$$E_A(\eta) = MMSE(H_A, \eta)$$
$$E_R(\eta) = MMSE(H_R, \eta) \tag{4.15}$$

Eventually, the location η is added to the subset such that:

$$\eta^* = \arg\max_{\eta}(E_A(\eta) - E_R(\eta)) \tag{4.16}$$

The justification for this choice of cost function is to expand the expressive capability of locations currently selected (large $E_A(\eta)$) while also describing well locations that remain (small $E_R(\eta)$). The rationale is essentially the same as that of the *RaVaG-Y*$_2$ algorithm of Section 3.6.

4.4.1.2 Spatially Maximum Distortion

In this case, we seek locations such that, if sensors are placed at them, the maximum distortion over space adheres to a bound D (see eq. (4.3)). Our method again sequentially selects sensing locations until the distortion criterion of eq. (4.3) by virtue of eq. (4.4) is met. At each iteration, the location η is added to the subset such that:

$$\eta^{*} = \arg \max_{\eta \in H_R} \left(\frac{\min_{\boldsymbol{x} \in A} \left(\left\| \hat{c}_{H_A \cup \eta}(\boldsymbol{x}) \right\|_2^2 \right)}{\lambda_{\max}\left(\hat{G}_{H_A \cup \eta} \right)} \right) \tag{4.17}$$

where A is the set of test locations, H_A and H_R are defined as in the previous subsection, $\hat{\boldsymbol{c}}_{(\cdot)}$ and $\hat{G}_{(\cdot)}$ are estimates of quantities first defined in eq. (4.4), $\|\cdot\|_2$ is the Euclidean norm and $\lambda_{max}(\cdot)$ denotes the largest eigenvalue of the argument. The cost function selects a location such that a lower bound on the variance reduction achieved by sensors at selected locations is maximized. The rationale in this case is the same as that behind the *GreePo* algorithm of Section 3.8.

4.5 Evaluation

We tested our design methodology for the proactive management scenario with a range of synthetic data. The simulated evaluation setting was exactly the same as that used for the reactive management scenario in Section 3.9: zero mean uniform white noise was fed into a symmetric 2-D low pass filter. White Gaussian noise of mean zero was added to sensor samples in all cases, resulting in a Signal to Noise Ratio

(SNR) equal to 10. The spatial support for the realizations was a field of square shape and size 10^4 m^2. We considered $N = 1500$ candidate locations, spread uniformly over an observation field. The only difference with the setup of Section 3.9 was that here, the observation field excluded a small strip of width 10 m around the edges of the 100x100 m^2 spatial support of the realizations, in order to avoid edge effects on measured covariances.

4.5.1 Exploration

The first step was to estimate the covariance curve *C(d)* by sequentially placing exploratory sensors in the observation field. To accomplish this, we used our *GreeX* exploration algorithm of Figure 4.2, setting the convergence parameter δ to 10^{-6}. This roughly corresponded to 0.1% of the scale of the initial cross-validation error of eq. (4.14).

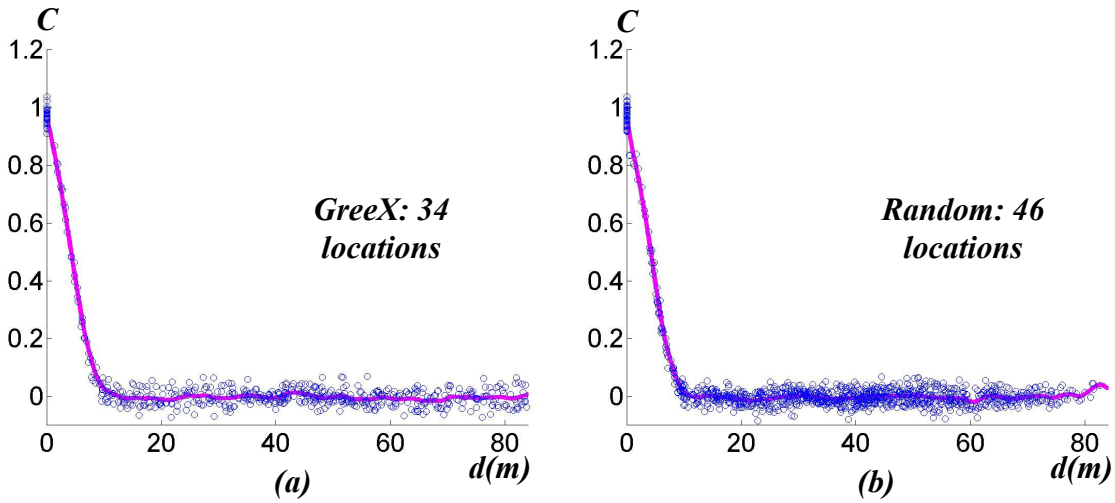

Figure 4.3: (a)*GreeX* exploration covariance curve (b)*Random* exploration covariance curve

Figure 4.3 shows the estimated covariance curves produced by *GreeX* and random exploration. It is well known that for isotropic (and generally for weakly stationary) random processes, the part of the covariance function closer to the origin has the most significant impact on interpolation [Ste99]. Intuitively, a good spatial interpolant at point $\boldsymbol{x}$ should depend mainly on observations near $\boldsymbol{x}$. It can be seen that both models capture the important region of the covariance curve which starts from distance 0 m and extends up to roughly 11 m.

However, in doing so, random exploration appoints 46 locations in total to place sensors at, while our *GreeX* exploration appoints 34 locations. *GreeX* thus achieves a 26% improvement in the number of exploratory sensors. To further illustrate the behavior of these exploration schemes, Figure 4.4 shows the values of the fitness function of eq. (4.13) after the addition of the final sensor into the exploratory subset.

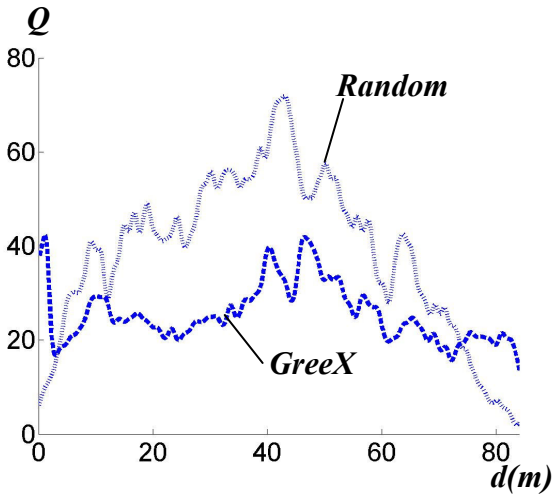

Figure 4.4: Fitness function

The fundamental difference of *GreeX* as compared to random exploration is its *waterfilling effect*: in *GreeX*, a sensor is added so as to maximally raise the fitness

function level at its most pronounced dips. This eventually results to data points being spread uniformly across the whole range of *d*. On the other hand, random exploration shows a high concentration of data points at distances around 42 m, which is a characteristic of the particular deployment and observation field used. Intuitively, random exploration would have to select many more locations in order to provide data values at any single argument distance far away from the characteristic one.

4.5.2 Design

The second step was deciding on actual locations to deploy sensors at, so as to perform interpolation. Specifically, we applied our greedy schemes of eq. (4.16) and (4.17) to obtain designs for criteria (4.2) and (4.3) respectively. Distortion was checked in the ensemble mean sense against *N* process realizations generated with the same procedure as the training data. In both the spatially averaged and the maximum case, the observation field *F* was discretized with a test grid of 80x80 locations and distortion was checked on this grid. To mitigate edge effects, locations within a small distance (~0.1m) from the edge of the observation field were disregarded for distortion calculation. The distortion target *D* was set to 0.5, which roughly corresponds to half the variance of the simulated process realizations at any point in space.

When applying our design schemes, we used two approximations. These were strictly targeted at reducing the computational burden of the design. Firstly, as per eq. (4.2) and (4.3), the algorithms should terminate when the estimated spatially averaged or maximum distortion respectively fall below the threshold *D*. To estimate distortion for each point on the test grid, we truncated the estimated covariance curves of Figure

4.3 to their respective *practical ranges*. According to geostatistical convention, the practical range is the distance where the value of covariance drops below 5% of its value at the origin [Dig06] (roughly 9 m in Figure 4.3). Intuitively, this means that sensors where considered to be correlated with locations within their practical range in a way quantified by the curves in Figure 4.3 and uncorrelated with all other locations. Secondly, for the maximum distortion heuristic of eq. (4.17) the minimum norm in the numerator was taken over the subset of possible design locations not yet selected in the design, instead of the actual test grid of 80x80 locations. This does not significantly impact the actual distortion performance of the design, because the initial set of possible locations $\boldsymbol{X}_0$ densely covered the observation field. In general, any such dense set of points would service the algorithm well.

Here, we are provided with a fixed set $\boldsymbol{X}_0$ of possible locations to deploy at. This is more realistic from a practical point of view and it generally precludes a regular grid design for example. We thus compared our design schemes with a completely random design:

Random design: Select k sensors at random to comprise the monitoring subset $\boldsymbol{X}_m$*. Keep increasing k until the spatially averaged distortion criterion (eq. (4.2)) or the maximum distortion criterion (eq. (4.3)) is satisfied.*

The random design was used in conjunction with the random exploratory curve of Figure 4.3(b) while both the greedy schemes of eq. (4.16) and (4.17) were used in conjunction with the *GreeX* curve of Figure 4.3(a).

Figure 4.5(a) shows the sizes of monitoring subsets obtained with random design and our schemes, for both distortion criteria (4.2) and (4.3). Figure 4.5(b) further shows the distortion performance achieved, if sensors are actually deployed at these locations. It can be seen that all schemes meet the distortion target D for the corresponding distortion criteria. In some cases, achieved distortion is in fact significantly smaller than the target. In these cases, the exploration curves of Figure 4.3, truncated at their practical ranges, are leading to overestimation of the actual distortion.

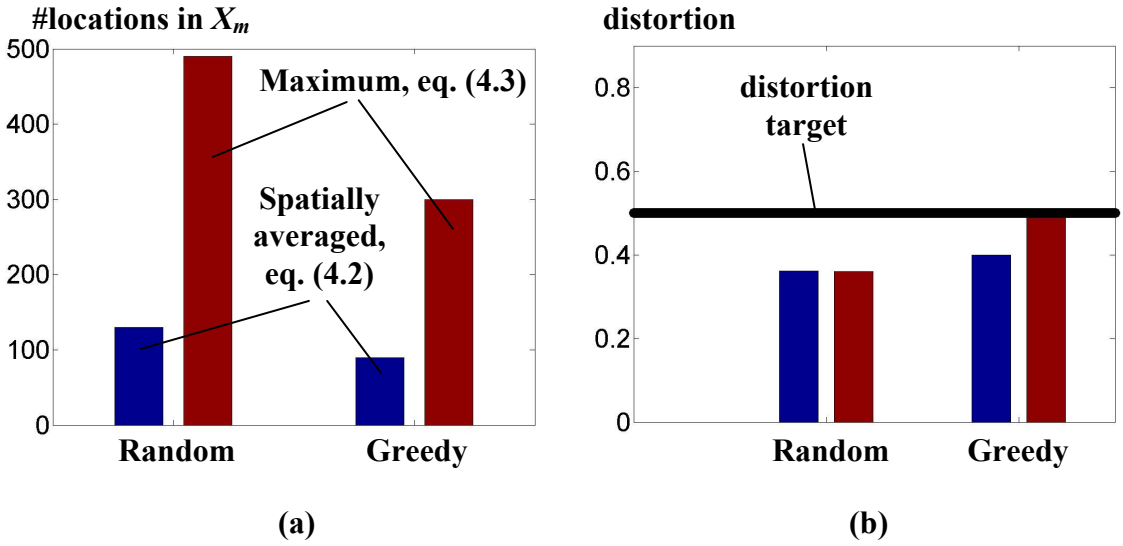

Figure 4.5: (a) Obtained designs, with vertical bars within the same scheme corresponding to different distortion objectives (b) Distortion performance

Our design schemes improve on random design in terms of total number of locations needed: for spatially averaged distortion, 130 monitoring locations are needed when designing randomly compared to 90 when designing with eq. (4.16), which represents a 30% improvement. In the case of maximum distortion, the improvement is 39% stemming from a total of 490 monitoring locations for random

design and 300 monitoring locations for greedy design based on eq. (4.17). Therefore, the schemes we propose here are overall a good choice for sensor network design driven by spatial interpolation applications, in the sense that they require a smaller total number of sensors.

4.6 Related Work

Extensive research on the design of monitoring networks has been carried out in the field of geostatistics [Dig06], [Ste99]. The basic difference of the geostatistical setting as compared to the sensor network one is that, in the former, repeated measurements at observation locations are generally not available, i.e., the time dimension cannot be exploited. Geostatistical methodologies only utilize a single measurement at each observation location to infer underlying statistical structure. Consequently, and in order to make mathematically sound inferences, they have to rely on assumptions about this structure. The prevalent choice is to assume that the physical phenomenon at any given set of monitoring sites follows a jointly Gaussian distribution with parameterized mean and covariance functions.

Based on this view, some authors describe locally optimal network designs. These are obtained by fixing the values of the mean and covariance parameters and finding monitoring locations that optimize an estimate of the MSE of interpolation. The parameter values can then be varied over a range and the best overall design, according to some criterion, kept [Zhu06], [Zim06]. Fully Bayesian methodologies have also been proposed which assume that a prior distribution on the model parameters is available. The prior can then be augmented through the use of data from

pre-existing designs. Working examples of such methods again use Gaussian processes [Dig06]. There is also a significant amount of work on designing a network especially to estimate values or priors for the model parameters [Bog99], [Zhu05], [Zim06]. Our approach on the other hand, does not assume a particular distribution on the data and performs exploration of the physical phenomenon as opposed to exploration of a parameter space. Additionally, this does not presume availability of past datasets, but rather is done in an online fashion: measurements gathered from sensors at already chosen exploratory locations affect choice of the next exploratory location.

In a sensor networks context, work related to ours appears in [Gue05]. Authors again utilize jointly Gaussian statistics and sequentially choose monitoring locations so as to maximize mutual information between already chosen and not yet chosen locations. However, their methodology assumes that the mean and covariance of the underlying Gaussian process model are known accurately enough beforehand, by means of a prior expert design. The same authors also present observation selection approaches for gradually learning a Gaussian process model [Kra07b]. Augmentation of their model is performed by maximizing the likelihood of observed data, based on a prior distribution over a predetermined set of parameter values. Initial parameter values and the exact distribution on them are themselves based on prior expert knowledge, which may not always be available. By contrast, in our work, the model relinquishes the need for data-dependent parameters in a great degree. Furthermore, unlike [Kra07b] our approach is also specifically applicable to min-max interpolation objectives.

Apart from the aforementioned greedy maximization of mutual information [Gue05], the design problem with a known covariance function has been addressed in geostatistical literature with greedy maximization of the kriging variance [Zim06] or with branch and bound maximization of the entropy [Ko05]. The last method, although exact, is practical only for a small number of monitoring locations while it additionally targets entropy. The shortcomings of using entropy as a guide to design are described in [Gue05]. The specific greedy heuristics we propose for this problem in Section 4.4 are distinct from those appearing in the literature.

Simulation has been routinely used in the context of Bayesian geostatistical design. Authors have simulated datasets according to Gaussian process models to statistically characterize interpolation error in the face of unknown model parameters [Ste99], [Zim06]. Others have proposed design approaches based solely on simulation [Dig06]. In Section 4.5 we have discussed a complimentary setup: regarding the second order statistics of the physical phenomenon as known, perform simulation on the deployment, i.e., the number and position of monitoring locations, as opposed to the interpolation model itself.

4.7 Summary

Our purpose in this chapter has been to address the problem of proactively managing the deployment of a wireless sensor network to support spatial interpolation. This is inherently a design problem and is of interest especially when the statistical structure of the underlying physical phenomenon is not known a-priori. We have presented a two-phase approach for design, applicable when the underlying covariance

function can be adequately modeled as an isotropic curve. First, through a sequential exploration scheme, which we called *GreeX*, this curve can be efficiently learned by online placement of sensors. Knowledge of the estimated statistics then allowed us to design the network to meet specific interpolation objectives. Simulation results indicate that our design methodology can provide significant gains in terms of required sensors when compared to simpler schemes.

4.8 Acknowledgements

This chapter is, in full, a reprint of material submitted for publication in the Proceedings of the IEEE Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2009 under the title “Designing Sensor Networks for Spatial Interpolation”. I was the primary researcher and author and Curt Schurgers supervised the research.

CHAPTER 5

5. PERSPECTIVES AND CONCLUSION

5.1 Background

In previous chapters we essentially studied and proposed solutions for a series of problems arising within the specific context of sampling and interpolation with a wireless sensor network (WSN). The network has in all cases been viewed as a distributed sampling system, measuring the values of a physical quantity in space and time. The goal was to obtain a continuous spatial estimate for the quantity of interest over time, i.e., even in areas where no sensors exist. The model employed for analysis was to regard the values measured at individual sensors as samples from realizations of a spatiotemporal random process, i.e., as random variables. We have seen that under the assumption of a spatial mean that is constant in time and ergodicity in time, the Mean Squared Error (MSE) based accuracy of such an estimate depends solely on the second order covariance structure of the random process.

Our goal in this chapter will be mainly to discuss possible extensions to the methods proposed previously and outline some additional 'uncharted' terrain regarding sampling-interpolation scenarios.

5.2 Sensing Model Generalizations

Often in real life scenarios, no knowledge on the second order covariance structure of the underlying phenomenon is available to us a-priori. In the reactive management setup of Chapter 3 this absence of statistical information was compensated by overdeploying the network to begin with, a step warranted by the low envisioned cost of sensors and similar practices in event detection scenarios. Then, during an initial learning phase, data were gathered from all sensors and were used to guide sensor selection and monitoring in a subsequent sensing topology management phase. Essentially, although there is no covariance model a-priori, through overdeployment, an abundant amount of data becomes available to the practitioner and can ultimately be used to estimate all necessary values of the covariance.

In this respect, the problem we addressed in Chapter 4, namely proactive management of the deployment, is much more demanding. In this case, the covariance function is still a-priori unknown in general, but data acquisition through sensor placement comes at a price: sensor placements ought to be done in an intelligent manner so that the covariance function is *efficiently* learnt online. The greatest difficulty posed by this situation is what interpolation model to use for inference on the covariance function itself so as to reliably capture the spatial correlations in measured data. Such inference should be capable of explicitly characterizing the effects of data addition to the covariance model either from a single or multiple sensors, without however relying on actual observed data because they are not yet available.

Chapter 4 introduced the use of local linear regression as a potential such model. The advantages of local linear regression are that it is largely non-parametric, since the bandwidth parameter h only loosely depends on observed data and its bias, i.e. the absolute difference of the estimate from the ground truth, is asymptotically independent from the pattern of supporting points in space. A first straightforward extension of our work, is to incorporate general, non-stationary covariance functions into the design methodology described here. Local linear regression is in fact directly extendable to the case of two spatial dimensions [Har04].

On the other hand, a fruitful direction for expansion in the reactive management scenario, is investigating the ability of the model to more directly exploit temporal correlations. So far we have seen how spatial correlation can be leveraged to reveal redundancy in a sampling-interpolation sensor network. The reason for this has been that the behavior of phenomena monitored in interpolation scenarios is known to be cohesive in space, especially at the scale of envisioned sensor network deployments. Many real world scenarios however also include attributes that evolve over time. If, for example the temporal evolution of values measured at sensors is highly predictable, we could take advantage of past measurements in addition to spatial correlations and employ a decreasing number of sensors over time.

At present, our model related assumptions entail spatial means and a spatial covariance structure that are constant in time (see eq. (2.5)-(2.8)). Time evolving properties can be detected by repeating the learning phase after fixed time intervals. A first possibility for experimentation is to forfeit the constant mean assumption and fit a linear or cubic trend to the time series obtained at each sensor during the learning

phase. This trend will then be subtracted from instantaneous sensor measurements and residuals will be used to explain spatial correlations. Linear time trends have been employed for the explanation of meteorological data [Le06], [Pac03], but the degree to which they preserve the spatial covariance structure, if at all, is still an open question. Note that, in the case of subtracted temporal trends, the resulting spatial covariance may generally be changing in time, although in an implicit manner.

A second possibility would be to assume an explicit dynamic model to describe time evolution of the spatial covariance. Dynamic behavior is often captured in the literature through Gauss-Markov models combined with Kalman filters [Des04]. However, a characteristic that requires special consideration here is that the input measurements for updating the state of the system will be potentially generated by a small set of sensors at each step, i.e., by the current monitoring subset. The challenge will then be to make correct inferences about the covariance structure over the whole observation field whilst relying only on few spatial locations.

Overall, the models and methods presented in the previous chapters are still appropriate in a wide range of situations. It should additionally be emphasized that enhancements described in this section are aimed more at completeness of the framework, without necessarily avoiding complexity.

5.3 Network Model Generalizations

Apart from issues relevant to modeling of sensor measurements, there is also a number of interesting network modeling aspects to our setup that have yet to be investigated. Specifically, for the problem of reactive management of Chapter 3, we

have proposed subset construction algorithms with the objective of extracting acceptable distortion performance from each individual subset while maximizing the number of subsets. Each subset thus contains a minimum number of sensors, in the sense that, if any one of the sensors comprising it depletes its energy reserves, the subset can no longer achieve the desired interpolation fidelity. However, during actual network operation, and under realistic circumstances, the initial battery energy as well as the rate of battery energy absorption may well differ among individual sensors. Patterns of individual sensor shutdowns have in fact been modeled as Markov processes [Kar05].

In this case the strict notion of a ‘subset’ of sensors becomes vague: a subset can contain sensors that are still operational, even after it cannot support the target distortion as a whole any longer. Since this will be occurring for all subsets, schemes can be envisaged which will dynamically and efficiently combine residual sensors from all ‘failed’ subsets to create new ones. An immediate consequence of a formulation that takes sensor shutdowns into account, would be a gradual degradation of the distortion performance of the network as a combined sensing entity, instead of the strict ‘hit-or-miss’ measure of performance that has been pursued so far. The rate of decay of the performance and its interaction with various factors would then be an interesting subject of investigation.

5.4 Performance Analysis

As has been mentioned quite frequently in the previous chapters, the problem of subset construction is inherently combinatorial. The performance analysis of solutions

proposed, and especially of the *RaVaG* algorithms which have shown the most promise, should rely on manipulation of expressions involving MSEs, i.e., of eq. (3.21). To date there are very few theoretical analyses even of the generic MSE expression of eq. (2.12), when the elements of the Grammian matrix are generated by observed data or by ideal models describing such data. In our case for example, the Grammian of all sensor vectors G_0, is ideally populated by evaluations of the covariance function of a space-time random process (as per eq. (2.11)); in actuality, it is populated only by estimates of such values (e.g. as per eq. (3.9)).

The reason behind this gap in the literature can be found in the highly non-linear behavior of the matrix inverse G_0^{-1}. The mathematical tools currently available to attack the problem involve perturbation theory [Das08a]. Very recently, a novel view of the covariance structure as a graph appeared in the literature [Das08a]. Translated in terms of our particular problem setup, vertices of the graph would be the sensor nodes and edges would exist among correlated sensors, bearing weights equal to respective correlations, as defined by the inner product (2.8). Based on this view and utilizing various perturbation bounds, analytical results involving the matrix inverse term (which was also called variance reduction in Section 3.8) have been derived for particular cases of graph structures [Das08a]. This demonstrates the usefulness of the graph view of covariance.

As discussed in the related work of Chapter 3, our particular problem is even more intricate than the one examined by these authors, since all sensors are random variables to be approximated simultaneously. Furthermore, our early experimentation has shown that, for our particular datasets, either the constraints imposed by the

perturbation analysis are too stringent or the bounds eventually obtained are too loose. This may call for the development of even more sophisticated tools to facilitate the analysis of algorithms such as *RaVaG*. The first step towards such a direction would be to initially assume very simple structures for the covariance of the underlying random process.

5.5 Multi-pass Deployments

Implicitly, our exposition of spatial interpolation scenarios for WSNs has so far focused on two extreme cases: 'one pass' deployments and 'infinitely controllable' deployments. These two extremes can in fact be relaxed under the unifying scope of *multiple pass deployments*. To elucidate this, consider the case where we have to deploy a sensor network in order to monitor a physical quantity over the entire observation field. The only information available to us before deployment is two bounds on the total number of sensors necessary: an absolute upper bound and a corresponding absolute lower bound, meaning that the necessary number of sensors definitely exceeds the lower bound but not the upper. If only a single pass may be used for deployment, a sensible solution would be to deploy the maximum possible number of sensors from the beginning and then impose reactive management on them, as described in Chapter 3. If, on the other hand, unlimited passes may be used, then sequential exploration and design can be employed instead, as described in Chapter 4.

However, if exactly two passes may be utilized (e.g. we can only pay for airplane fuel twice) and since efficient use of resources is the foremost objective, a sensible approach would perhaps dictate to initially deploy only the minimum possible

number of sensors. At that point, data from these sensors could be used to infer a tighter upper bound on the total number of sensors necessary. Given the luxury of more than two deployments, even more such inferences could be furnished to further tighten the upper and lower bounds and lead to fewer total sensors deployed eventually. The key idea behind such inferences is to see how much additional information these intermediate deployments can provide about the covariance function. In other words, multi-pass deployments may potentially be used to predict the quality of covariance function estimation before the actual estimation takes place.

5.6 Conclusion

The vision for networks of wireless microsensors has been rapidly gaining proportions in the last decade. The scope of envisioned applications is vast and thus necessitates a diverse set of techniques. This spectrum can be roughly categorized into two different types of relevant applications, namely event detection and spatiotemporal sampling and interpolation. These differ fundamentally in terms of how correlations among sensor readings can be characterized and interpreted.

Our research focus has specifically been on sampling-interpolation type scenarios. The ultimate goal was to identify and exploit inter-sensor correlations in this setting so as to efficiently achieve desired interpolation accuracy. In lieu of assuming a particular probability distribution for the measured values, we have relied on the generic Hilbert space of second order random variables, each corresponding to a single sensor. Within this model, best linear interpolation can be expressed as an orthogonal projection and the associated Mean Squared Error depends only on a

spatial covariance function. This function essentially encompasses all information about sensor interdependencies.

We then examined the class of scenarios where we have to manage sampling and interpolation *reactively* in an already (over)deployed sensor network. The strategy we proposed was to devise subsets of sensors, each individually capable of producing data to support the interpolation task. Specific algorithms thereon have shown substantial improvements in terms of the number of data producing sensors and energy expended to convey these data to a central processing location. Additionally, we pursued solutions to cater for two distinct notions of interpolation quality: spatially averaged and spatially maximum MSE.

Another class of scenarios we examined was that of proactive network deployment: how to deploy a minimum number of sensors from scratch in order to meet desired interpolation objectives. The path we proposed was to first explore the covariance space and then place sensors according to the estimated covariance curve. Described algorithms have succeeded in producing placements that provide accurate enough interpolations using far fewer sensors than simpler design schemes.

Perhaps a key methodological aspect of our work has been that, in a sensor network context, a collection of infinite dimensional random variables, such as temperature readings, can be approximated by finite length time series measured at the sensor locations. The associated Hilbert space quantities and operators can then be approximated through algebraic, i.e., matrix formulations and this development suggests a strong potential for impacting the field, even beyond the scope of the particular problems discussed here.

REFERENCES

[Abr04] Abrams, Z., Goel, A. and Plotkin, S., "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks", in IPSN 2004.

[Ai07] Ai, X., Srinivasan, V. and Tham, C.-K., "DRACo: Distributed, Robust and Asynchronous Coverage in Wireless Sensor Networks", in SECON 2007.

[Ala87] Alabert, F., "The Practice of Fast Conditional Simulations through the LU Decomposition of the Covariance Matrix", in Mathematical Geology, Vol. 19, No. 5, 1987.

[Aro05] Arora, A., "ExScal: Elements of an Extreme Scale Wireless Sensor Network", in RTCSA 2005.

[Asp04] D'Aspremont, A., Ghaoui, E. L., Jordan, M. I. and Lanckriet, G. R. G., "A direct formulation for sparse PCA using semidefinite programming", Technical Report CSD-04-1330, Division of Computer Science, University of California, Berkeley.

[Baj06] Bajwa, W., Haupt, J., Sayeed, A., and Nowak, R., 'Compressive Wireless Sensing', in IPSN 2006.

[Bal07] Balzano, L. and Nowak, R., "Blind Calibration of Sensor Networks", in IPSN 2007.

[Bap05] Bapat, S., Kulathunami, V. and Arora, A., "Analyzing the Yield of ExScal, a Large-Scale Wireless Sensor Network Experiment", in ICNP 2005.

[Ber01] Berger, J. O., Oliveira, V. D., Sanso, B., "Objective Bayesian Analysis of Spatially Correlated Data", in Journal of The American Statistical Association, Vol. 96, No. 456, 2001, p. 1361-1374.

[Beu70] Beutler, F. J., "Alias-Free Randomly Timed Sampling of Stochastic Processes", in IEEE Transaction on Information Theory, Vol. 16, No. 2, March 1970.

[Bia02] Biagioni, E. and Bridges, K., "The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species", in International Journal of High Performance Computing Applications, Vol. 16, No. 3, Fall 2002.

[Bog99] "Optimal Spatial Sampling Design for the Estimation of the Variogram Based on a Least Squares Approach", in Water Resources Research, Vol. 35, No. 4, p. 1275-1289, April 1999.

[Bow97] Bowman, A. W. and Azzalini, A., "Applied Smoothing Techniques for Data Analysis", Oxford University Press, 1997.

[Car04] Carbunar B., Grama A., Vitek J. and Carbunar O., "Coverage Preserving Redundancy Elimination in Sensor Networks", in SECON 2004.

[Con05] http://www.controldesign.com/industrynews/2005/040.html, "Wireless sensor networks market expected to skyrocket".

[Cot02] Cotter, S. F., Kreutz-Delgado, K. and Rao, B. D., "Efficient Backward Elimination Algorithm for Sparse Signal Representation Using Overcomplete Dictionaries", in IEEE Signal Processing Letters, Vol. 9, No. 5, May 2002.

[Cot05] Cotter, S. F., Bhaskar, B. D., Engan, K. and Kreutz-Delgado, K., "Sparse Solutions to Linear Inverse Problems with Multiple Measurement Vectors", in IEEE Transactions on Signal Processing, Vol. 53, No. 7, July 2005.

[Cra67] Cramer, H. and Leadbetter, M. R., "Stationary and Related Stochastic Processes: Sample Function Properties and Their Applications", Wiley, 1967.

[Cul04] Culler, D., Estrin, D. and Srivastava M., "Guest Editors' Introduction: Overview of Sensor Networks", in Computer, Vol. 37, No. 8, August 2004.

[Das08a] Das, A. and Kempe, D, 'Algorithms for Subset Selection in Linear Regression', in STOC 2008.

[Das08b] Das, A. and Kempe, D, "Sensor Selection for Minimizing Worst-Case Prediction Error", in IPSN 2008.

[Dav97] Davis, G., Mallat, S. and Avellaneda, M., "Adaptive Greedy Approximations", in Journal of Constructive Approximation, Vol. 13, p. 57-98, 1997.

[Deg06] Degesys, J., Rose, I., Patel, A. and Nagpal, R., "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks", in IPSN 2006.

[Des04] Deshpande, A., Guestrin, C., Madden, S. R., Hellerstein, J. M. and Hong, W., "Model-Driven Data Acquisition in Sensor Networks", in VLDB 2004.

[Deu01] Deutsch, F., "Best Approximation in Inner Product Spaces", Springer-Verlag, 2001.

[Dig06] Diggle, P. J. and Ribeiro, P. J. J., "Model-based Geostatistics", Springer 2006.

[Don03] Donoho, D. L. and Elad, M., "Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization", in PNAS 2003.

[Don04] Dong, M., Tong, L. and Sadler, B. M., "Effect of MAC Design on Source Estimation in Dense Sensor Networks", in ICASSP 2004.

[Dua06] Duarte, M. F., Wakin, M. B., Baron, D. and Baraniuk, R. G., "Universal Distributed Sensing via Random Projections", in IPSN 2006.

[Dut05] Dutta, P., Grimmer, M., Arora, A., Bibyk, S. and Culler, D., "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events", in IPSN 2005.

[ExS04] http://www.cast.cse.ohio-state.edu/exscal/, "Exscal: Extreme Scale Wireless Sensor Networking".

[Fue07] Fuentes, M., Chaudhuri, A. and Holland D. M., "Bayesian Entropy for Spatial Sampling Design of Environmental Data", in Enviromental and Ecological Statistics, Vol. 14, No. 3, September 2007.

[Gue04] Guestrin, C., Bodic, P., Thibaux, R., Paskin, M. and Madden, S., "Distributed Regression: An Efficient Framework for Modeling Sensor Network Data", in IPSN 2004.

[Gue05] Guestrin, C., Krause, A. and Singh, A. P., "Near Optimal Sensor Placements in Gaussian Processes", in ICMP 2005.

[Har04] Hardle, W., Muller, M., Sperlich, S. and Werwatz, A., "Nonparametric and Semiparametric Models", Springer 2004.

[Har06] Harmon, T. C., Ambrose, R. F., Gilbert, R. M., Fisher, J. C., Stealey, M. and Kaiser, W. J., "High Resolution River Hydraulic and Water

Quality Characterization Using Rapidly Deployable Networked Infomechanical Systems (NIMS RD)", Technical Report No. 60, Center for Embedded Networked Sensing (CENS).

[Has90] Hastie, T., J. and Tibshirani, R. J., "Generalized Additive Models", Chapman & Hall/CRC Press, 1990.

[Hog07] http://www.hogthrob.dk/, Hogthrob Networked on-a-chip nodes for sow monitoring.

[Hu08] Hu, S. and Motani, M., "Early Overhearing Avoidance in Wireless Sensor Networks", in NETWORKING 2008, LNCS 4982, p. 26–35, 2008.

[Hua03] Huang, C.-F. Tseng, Y.-C., "The Coverage Problem in a Wireless Sensor Network", in WSNA 2003.

[Int04] http://techresearch.intel.com/articles/Exploratory/1501.htm, Sensor Networks Research.

[Jin06] Jindal, A. and Psounis, K., "Modeling Spatially Correlated Data in Sensor Networks", in ACM Transactions on Sensor Networks, Vol. 2, No. 4, November 2006, p. 466-499.

[Jua02] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. and Rubenstein, D. "Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet", in ASPLOS-X 2002.

[Kar05] Kar, K., Krishnamurthy, A. and Jaggi, N., "Dynamic Node Configurations in Networks of Rechargeable Sensors", in INFOCOM 2005.

[Ken98] Kendall, S. W. and Lieshout van M. N. M., "Stochastic Geometry: Likelihood and Computation", Chapman & Hall/CRC Press, 1998.

[Kin06] Kini, A. V., Veeraraghavan, V., Singhal, N. and Weber, S., "SmartGossip : An Improved Randomized Broadcast Protocol for Sensor Networks", in IPSN 2006.

[Ko05] Ko, C., Lee, J. and Queyranne, M., "An Exact Algorithm for Maximum Entropy Sampling", in Operations Research, Vol. 43, No. 4., p. 684-691.

[Kou06] Koushanfar, F., Taft, N. and Potkonjak, M., "Sleeping Coordination for Comprehensive Sensing Using Isotonic Regression and Domatic Partitions", in Infocom 2006.

[Kra06] Krause, A., Guestrin, C., Gupta, A. and Kleinberg, J., "Near-optimal Sensor Placements: Maximizing Information while Minimizing Communication Cost", in IPSN 2006.

[Kra07a] Krause, A., McMahan, B. H., Guestrin, C. and Gupta, A., "Selecting Observations Against Adversarial Objectives", in NIPS 2007.

[Kra07b] Krause, A. and Guestrin, C., "Nonmyopic Active Learning of Gaussian Processes: An Exploration – Exploitation Approach", CMU Technical Report, CMU-ML-07-105, 2007.

[Kri05] Krishnamachari, B., 'Networking Wireless Sensors', Cambridge University Press, 2005.

[Kum04] Kumar, S., Lai, T. H. and Balogh, J., "On k-coverage in a Mostly Sleeping Sensor Network", in MobiCom 2004.

[Kun03] Kunsch, H. R., Agrell, E. and Hamprecht, F. A., "Optimal Lattices for Interpolation of Stationary Random Fields", Research Report No. 119, ETH, 2003.

[Lan03] Langendoen, K. and Reijers, N., "Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison", in Computer Networks, 2003.

[Le06] Le, N. D. and Zidek, J. V. , "Statistical Analysis of Environmental Space-Time Processes", Springer, 2006.

[Ler02] Lerner, U., "Hybrid Bayesian Networks for Reasoning about Complex Systems", PhD Dissertation, Stanford University, 2002.

[Li08] Li, Y., Thai, M. T. and Wu, W. (Eds.), "Wireless Sensor Networks and Applications", Springer, 2008.

[Lia06] Liaskovitis, P. and Schurgers, C. "A Distortion Aware Scheduling Approach for Wireless Sensor Networks", in DCOSS 2006.

[Lia07a] Liaskovitis, P. and Schurgers, C. "Leveraging Redundancy in Sampling-Interpolation Applications for Sensor Networks", in DCOSS 2007.

[Lia07b] Liaskovitis, P. and Schurgers, C. “Leveraging Redundancy in Sampling Interpolation Applications for Sensor Networks: A Spectral Approach”, submitted to ACM Transactions on Sensor Networks, 2007.

[Lim90] Lim, J. S. “Two Dimensional Signal and Image Processing”, Prentice Hall, 1990.

[Mai02] Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D. and Anderson, J., “Wireless Sensor Networks for Habitat Monitoring”, in WSNA 2002.

[Mar01] Marvasti, F., Editor, “Nonuniform Sampling Theory and Practice”, Kluwer Academic Publishers, 2001.

[Mil02] Miller, A. ‘Subset Selection in Regression’, Chapman & Hall/CRC Press, 2002.

[Mog05] Moghaddam, B., Weiss, Y. and Avidan, S., “Spectral Bounds for Sparse PCA: Exact and Greedy Algorithms”, in Advances in Neural Information Processing Systems 18, MIT Press, 2006.

[Mol03] Moller, J. and Waagepetersen, R. P., “Statistical Inference and Simulation for Spatial Point Processes”, Chapman & Hall/CRC Press, 2003.

[Moo00] Moon, T. K. and Stirling, W. C., “Mathematical Methods and Algorithms for Signal Processing”, Prentice Hall, 2000.

[Not02] Nott, D. J. and Dunsmuir, W. T. M., “Estimation of non-stationary spatial covariance structure”, in Biometrika, Vol. 89, p. 819-829, 2002.

[Oya06] Oya, A., Navarro-Moreno, J. and Ruiz-Molina, J. C., “Spatial Random Field Simulation by a Numerical Series Representation”, in Stochastic Environmental Research and Risk Assessment, Vol. 21, No. 4, April 2007.

[Pac03] Paciorek, C., “Nonstationary Gaussian Processes for Regression and Spatial Modeling”, PhD Dissertation, Carnegie Mellon University, May 2003.

[Pap98] Papadimitriou C. H. and Steiglitz K., “Combinatorial Optimization - Algorithms and Complexity”, Dover Publications 1998.

[Pat04] Patwari, N. and Hero, A. O., “Manifold Learning Algorithms for Localization in Wireless Sensor Networks”, in ICASSP 2004.

[Per04] Perillo, M., Ignjatovic, Z., and Heinzelman, W., “An Energy Conservation Method for Wireless Sensor Networks Employing a Blue Noise Spatial Sampling Technique”, in IPSN 2004.

[Rag02] Raghunathan, V., Schurgers, C., Park, S. and Srivastava M. B., "Energy-Aware Wireless Microsensor Networks," in IEEE Signal Processing Magazine, Vol.19, No.2, p. 40-50, March 2002.

[Rao03] Rao, B. D, Engan, K., Cotter, S. F., Palmer, J. and Kreutz-Delgado, K., “Subset Selection in Noise Based on Diversity Measure Minimization”, in IEEE Transactions on Signal Processing, Vol. 51, No. 3, March 2003.

[Ras06] Rasmussen, C. E. and Williams, C. K. I., “Gaussian Processes for Machine Learning”, MIT Press, 2006.

[Sch05] Schäfer, J. and Strimmer, K., “A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics”, in Statistical Applications in Genetics and Molecular Biology, Vol. 4, Issue 1, 2005.

[Sch02] Schurgers, Curt, “Energy-aware Communication Systems”, PhD Dissertation, University of California, Los Angeles, 2002.

[Sen07] http://sensorscope.epfl.ch/index.php/Main_Page, Sensorscope Wireless Distributed Sensing System for Environmental Monitoring.

[Sim06] Similä, T. and Tikka, J., “Common Subset Selection of Inputs in Multiresponse Regression”, in International Joint Conference on Neural Networks, 2006.

[Sli01] Slijepcevic, S. and Potkonjak, M., “Power Efficient Organization of Wireless Sensor Networks”, in ICC 2001.

[Sma03] Small, T. and Haas, Z. J., “The Shared Wireless Infostation Model - a New Ad Hoc Networking Paradigm”, in MOBIHOC 2003.

[Sta02] Stark, H. and Woods, J. W., “Probability and Random Processes with Applications to Signal Processing”, Prentice Hall, 2002.

[Ste99] Stein M. L., “Interpolation of Spatial Data: Some Theory for Kriging”, Springer 1999.

[Sto01] Stoica, P. and Moses, R. “Spectral Analysis of Signals”, Prentice Hall, 2005.

[Sun05] Sung, Y., Tong, L. and Poor, H. V., “Sensor Activation and Scheduling for Field Detection in Large Sensor Arrays”, in IPSN 2005.

[Sze04a] Szewczyk, R., Mainwaring, A., Polastre, J. and Culler, D, “An Analysis of a Large Scale Habitat Monitoring Application”, in SenSys 2004.

[Sze04b] Szewczyk, R., Polastre, J., Mainwaring, A. and Culler, D, “Lessons from a Sensor Network Expedition”, in EWSN 2004.

[Tho01] Thompson, D. J. “Multitaper Analysis of Nonstationary and Nonlinear Time Series Data”, in Nonlinear and Nonstationary Signal Processing, Cambridge University Press, 2001.

[Tro06] Tropp, J. A., Gilbert, A. C. and Strauss, M. J., “Algorithms for Simultaneous Sparse Approximation Part I: Greedy Pursuit”, in Signal Processing, Vol. 86, No. 3, March 2006, p. 572-588.

[Vur06] Vuran, M. C. and Akyildiz, I. F., “Spatial Correlation Based Collaborative Medium Access Control in Wireless Sensor Networks”, in IEEE/ACM Transactions on Networking, Vol. 14, No. 2, April 2006, p. 316 -329.

[Wae99] Waele, S. and Broersen, P. M. T., “Reliable LDA-Spectra by Resampling and ARMA-Modeling”, in IEEE Transactions on Instrumentation and Measurement, Vol. 48, No. 6, Dec. 1999.

[Wan95] Wand, M. P. and Jones, M. C., “Kernel Smoothing”, Chapman & Hall/CRC Press, 1995.

[Wan03] Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R. and Gill, C., “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks”, in SenSys 2003.

[Wan07] Wang, W., Garofalakis, M. and Ramchandran, K., “Sparse Random Projections for Refinable Approximation”, in IPSN 2007.

[Wie05] Wiens, D. P., “Robustness in spatial studies II: minimax design”, in Environmetrics 2005, Vol 16, p. 205-217.

[Wip04] Wipf, D. and Rao, B. D., “Sparse Bayesian Learning for Basis Selection”, in IEEE Transactions on Signal Processing, Vol. 52, No. 8, August 2004, p. 2153-2164.

[Wip07] Wipf, D. and Rao, B. D., "An Empirical Bayesian Strategy for Solving the Simultaneous Sparse Approximation Problem", in IEEE Transactions on Signal Processing, Vol. 55, No. 7, July 2007, p. 3704-3716.

[Yu04] Yu, Y., Estrin, D., Rahimi, M. and Govindan, R., "Using more realistic data models to evaluate sensor network data processing algorithms", in LCN 2004.

[Zha03] Zhao, Q. and Tong, L., "Quality-Of-Service Specific Information Retrieval for Densely Deployed Sensor Networks", in MILCOM 2003.

[Zha05] Zhang, H. and Hou, J. C., "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks", in Ad Hoc and Sensor Wireless Networks, March 2005.

[Zhu05] Zhu, Z. and Stein, M. L., "Spatial Sampling Design for Parameter Estimation of the Covariance Function", in Journal of Statistical Planning and Inference, Vol. 134, No. 2, October 2005, p. 583-603.

[Zhu06] Zhu, Z. and Stein, M. L., "Spatial Sampling Design for Prediction with Estimated Parameters", in Journal of Agricultural, Biological, and Environmental Statistics, Vol. 11, No. 1, p. 24-44.

[Zim06] Zimmerman, D. L., "Optimal network design for spatial prediction, covariance parameter estimation and empirical prediction", in Environmetrics 2006, Vol. 17, p. 635-652.

[Zou06] Zou, H., Hastie, T. and Tibshirani, R., "Sparse Principal Component Analysis", in Journal of Computational and Graphical Statistics, Vol. 15, No. 2, 2006, p. 262-286.