

UNIVERSITY OF CALIFORNIA  
Los Angeles

Cooperation Utility in Sensing

A dissertation submitted in partial satisfaction of the  
requirements for the degree Doctor of Philosophy  
in Electrical Engineering

by

Yu-Ching Tong

2009

©Copyright by

Yu-Ching Tong

2009

The dissertation of Yu-Ching Tong is approved:

---

Kung Yao

---

Lieven Vandenberghe

---

Mario Gerla

---

Gregory Pottie, Committe Chair

University of California, Los Angeles

2009

To my parents,  
my aunt,  
and my sister for their support.

# Contents

<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>4</b>
<b>3 Uniform Contribution</b>	<b>9</b>
3.1 Introduction . . . . .	9
3.2 Uniform utility and its consequence . . . . .	10
3.3 Localization case study . . . . .	13
3.3.1 Observation uncertainty model . . . . .	13
3.3.2 Localization uncertainty . . . . .	14
3.3.3 Localization Utility Simulation . . . . .	17
3.3.4 Localization Conclusion . . . . .	20
<b>4 Non-Uniform Utility Metric</b>	<b>23</b>
4.1 Introduction . . . . .	23
4.2 Sensor Utility Statistics . . . . .	24
4.2.1 Order Statistics . . . . .	24
4.2.2 Uniform Disk . . . . .	25
4.2.3 2D Gaussian . . . . .	25
4.2.4 Expected Utility by varying $k, n$ . . . . .	27
4.3 Cooperation . . . . .	30
4.4 Very large number of sensors . . . . .	34
4.5 Non-uniform utility conclusion . . . . .	36
<b>5 Coverage</b>	<b>38</b>
5.1 Introduction . . . . .	38
5.2 Prior art on coverage . . . . .	39
5.3 Sensor model and utility metric . . . . .	40
5.4 Regular deployment . . . . .	41

5.4.1	Independent sensors . . . . .	42
5.4.2	Local cooperation sensors . . . . .	43
5.4.3	Large scale of cooperation between sensors . . . . .	45
5.5	Random deployment . . . . .	49
5.6	Conclusion . . . . .	52
<b>6</b>	<b>Reconstruction</b>	<b>53</b>
6.1	Introduction . . . . .	53
6.2	Source and Classical reconstruction technique . . . . .	56
6.2.1	Nyquist sampling . . . . .	56
6.2.2	Nearest sample . . . . .	57
6.2.3	Linear Interpolation . . . . .	57
6.2.4	Spline Interpolation . . . . .	57
6.2.5	Performance comparison . . . . .	58
6.3	Modifying the source . . . . .	60
6.3.1	Determine where the breaks are . . . . .	61
6.3.2	Finding the breaks in the first place . . . . .	62
6.3.3	Simulation result . . . . .	63
6.4	Trade off of sample size and reconstruction complexity . . . . .	64
6.4.1	Wavelet Compression . . . . .	65
6.4.2	Compressed Sensing . . . . .	68
6.4.3	Comparing Compressed Sensing with classical technique . . . . .	74
6.4.4	Comparing Compressed Sensing with classical technique with modified model, using wavelet representation as reference . . . . .	75
6.5	Conclusion . . . . .	78
<b>7</b>	<b>Conclusion</b>	<b>79</b>
	<b>Bibliography</b>	<b>82</b>

# List of Figures

3.1	Prior utility rate to the total utility . . . . .	12
3.2	Prior utility rate to the total utility with random sensor utility . . . . .	12
3.3	One realization of of localization utility . . . . .	21
3.4	Number of sensors to achieve 90% of total utility . . . . .	22
4.1	Simulation and theoretical order statistic of distance to origin, uniform disk	26
4.2	Simulation and theoretical order statistic of distance to origin, planar normal	27
4.3	Nearest sensor expected utility evolution from $n$ to $n+1$ in a disk . . . . .	28
4.4	Nearest sensor expected utility evolution from $n$ to $n+1$ in a 2D normal distribution . . . . .	28
4.5	Relative utility in disk . . . . .	29
4.6	Relative utility in 2D normal distribution . . . . .	29
4.7	Fading outage in a disk . . . . .	31
4.8	Fading outage in 2D normal distribution . . . . .	32
4.9	Fading outage in a disk, fixed goal . . . . .	33
4.10	Fading outage in 2D normal distribution, fixed goal . . . . .	33
4.11	Expected aggregated utility from constant density in an annulus . . . . .	36
4.12	PDF of the nearest sensor distance . . . . .	37
5.1	Sensor in Lattice example . . . . .	41
5.2	Complete coverage with independent sensors . . . . .	42
5.3	Sensors surround origin . . . . .	43
5.4	Diminishing return of cooperation . . . . .	45
5.5	The closest point of the hexagon to the origin . . . . .	46
5.6	Example of deployment, no cooperation . . . . .	50
5.7	Utility - Cooperation - Sensors count . . . . .	50
6.1	Reconstruction example . . . . .	59
6.2	Performance comparison between reconstruction method and resources trade-off . . . . .	59
6.3	Break point example . . . . .	60
6.4	Rate of meeting the QoS for a given initial sampling rate . . . . .	64

6.5	Outage rate to meeting QoS for a given initial sampling rate . . . . .	65
6.6	Fourier Transform domain . . . . .	67
6.7	Wavelet Transform domain . . . . .	67
6.8	Sparse in frequency domain . . . . .	73
6.9	Sparse frequency with discrete discontinuities . . . . .	73
6.10	Comparing compressed sensing with classical reconstruction . . . . .	74
6.11	Comparing compressed sensing with classical reconstruction and steps in signal	76
6.12	Reconstruction examples, Spline with EM and CS . . . . .	77
6.13	Comparing compressed sensing with classical reconstruction and steps in sig- nal, high signal frequency . . . . .	77

# List of Tables

5.1	Number of sensors and the ring distance multiplier . . . . .	44
5.2	Sensor Location in lattice . . . . .	47
5.3	Numerical example of cooperation size and deployment density trade off to achieve 95% coverage . . . . .	51
5.4	Numerical example of cooperation size and deployment density trade off to achieve 90% coverage . . . . .	51

## ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Professor Greg Pottie, for his guidance and enlightenment. Through our various topic discussions, I had learned much from Greg, lessons which I can also apply in the years to come. I would also like to thank Professor Kung Yao, Professor Lieven Vandenberghe, and Professor Mario Gerla for taking the time to serve on my committee. My appreciation also goes to Hugh Luo, Ameesh Pandya, Nabil Hajj Chehade, Kathy Kong, Kevin Ni, Yue Zhao and many others for their ideas and time.

On the professional front, I would like to thank my former colleagues at Boeing for their support and understanding; especially to Ernie Astin, Elizabeth Klein-Lebbink, Chris Taji, Dave Ziers, the GOES N-P communication system engineering team and the comm. system I&T team. I also would like to thank the Learning Together Program at Boeing.

Finally on the home front, I want to thank my parents, Chi Pan Tong and Kin Sang Tong, and my sister, Ka-Yan Tong, M.D., for their confident in me. I am also forever in debt to my aunt, Sui Sang Luk, and my mom, whose logistical support made this work possible.

Chapter 3 and 4 is an expanded version of [91].

## VITA

- 1979      Born, Hong Kong
- 2001      B.S., Electrical Engineering  
            magna cum laude  
            University of California, Los Angeles  
            Los Angeles, California
- 2001 - 2007    Member of Technical Staff  
            Boeing Satellite System  
            El Segundo, California
- 2002      M.S., Electrical Engineering  
            University of California, Los Angeles  
            Los Angeles, California

## PUBLICATIONS

- H. Luo,    Y.-C. Tong, and G. Pottie, "A two-stage DPCM scheme for wireless sensor networks," *IEEE International Conference on Acoustic, Speech, and Signal Processing*, Philadelphia, PA, USA, March, 2005
- Y. Tong,    G. Pottie, "The marginal utility of cooperation in sensor networks", *Information Theory and Applications Workshop*, 2008

# ABSTRACT OF THE DISSERTATION

Cooperation Utility in Sensing

by

Yu-Ching Tong

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2009

Professor Gregory Pottie, Chair

We present arguments that a small number of sensors within the network provides most of the utility. That is, cooperation of more than a small number of nodes has little benefit. We present two scenarios. In the first scenario, all sensors provide identical utility, and their utilities are aggregated sequentially. The second scenario is sensor fusion with signal strength decreasing with distance. In that scenario the source is at the origin and the sensors are distributed, either uniformly or according to a planar standard normal distribution. We also vary the total number of sensors distributed in both scenarios to observe the utility/density trade off. Localization using the Fisher Information as the utility metric is used to demonstrate that few sensors are sufficient to derive most of the utility out of the sensor network. Simulation results back up an order statistics analysis of the behavior.

The implication is that while co-operation is useful for some objectives such as combating fading and uncertainty of individual sensors, it is inefficient as a mean to increase

the utility of a sensor network if the best sensor's utility is significantly short of the desired utility.

In addition, asymptotic results under fixed density are presented. In this situation, the utility improvement is logarithmic at best as the number of sensors and the distance to the sensors increase.

Coverage area as a utility metric is considered in a lattice deployment and a random deployment scenario. In both cases small neighborhood cooperation provides noticeable improvement.

In reconstruction problems the effectiveness of global versus local cooperation depends heavily on the model relating the measurement and the source. When the model describes the relationship accurately, reconstruction, regardless of cooperation size, will work well. This problem illustrates that the model has a larger impact than cooperation size.

# Chapter 1

## Introduction

In a sensor network, cooperation size among sensors impacts communication resources consumption. At the same time too few measurements may result in very complex computation problems or high uncertainty in the results. An appropriately sized cooperation neighborhood enables the sensor network to meet the desired sensing quality without consuming excessive communication or computing resources. We will see that a relatively small cooperation size is sufficient to provide noticeable benefits from cooperation.

In [1, 2, 3, 4] the authors discussed how to select a sensor that will provide the most utility among sensors based on the utility each sensor is expected to contribute. We will extend their observation to consider how the size of cooperation impacts the performance of these various strategies. We will see that the first few sensors provide the bulk of the utility. We will start off in chapter 3 considering the case where every sensor contributes the same utility to the sensing objective. In this case, we will see that although arbitrary utility can be achieved, the marginal utility from each individual sensor diminishes. Using localization

as an example, we will see that with a small number of sensors cooperating, selection of a good set of sensors is important. On the other hand when a large number of sensors are involved, overall utility improves only marginally from the small set. In addition, when a large numbers of sensors are involved, selection algorithms have little impact on the overall utility.

In chapter 4, we will consider a more realistic utility model where an individual sensor's utility falls off as a function of distance. From [5, 6] we learn that some level of cooperation can help mitigate channel degradations such as fading. Those results established the usefulness of small scale cooperation in a communication setting. We extended the cooperation size to larger neighborhoods, and see that cooperation does not provide any additional benefit over the small neighborhood. We will observe that under a fixed density condition, under most distance loss exponents the total utility of the sensor network is *bounded* even when we allow the total number of sensors to be arbitrarily large. This result reinforces the observation in chapter 3 that a few sensors dominate the overall sensor network performance for a particular source. We will also see that few sensors cooperating can help mitigate degradations such as fading effectively, but cooperation alone cannot be used to achieve any arbitrary quality of service goal.

A very popular problem in sensor networks is coverage of the network. There are various forms to the coverage problem, such as density of sensors necessary for a certain coverage rate [7, 8, 9], coverage under a multiple view requirement [10] and redundancy problems such as those studied in [11, 12]. Coverage inherently is typically a multi-sensor problem, but it does not necessary require explicit cooperation among sensors. In chapter

5 we will consider coverage as the utility objective and observe how small scale cooperation improves coverage. Small gaps of coverage can be filled by cooperating with a few neighbors. On the other hand, large cooperation cannot be used to fill a large gap due to the bounded utility that can be achieved by cooperation, as we have seen in chapter 4.

Reconstruction from samples is another one of those problems that requires some form of cooperation. Many studies on this topic exist. The Nyquist [13] and Shannon [14] sampling theorem is one of the fundamental concepts in communication. In reconstruction problems, a model relating the samples and the source must be provided. We will compare several models and their corresponding reconstruction algorithms. Some of those models require all of the samples to be used to reconstruct the source, while other models reconstruct a small piece of the source with a few samples at a time. We will observe that with an accurate model a local reconstruction technique can be effective. We will also observe that with an inferior model, even using all of the measurements the global reconstruction technique will still result in poor reconstruction performance.

Thus, whether we consider sensing of a single source, coverage, or reconstruction, in this thesis we show that local small-scale cooperation can be useful, but large scale cooperation will not overcome an insufficient density of samples (or a poor model of the phenomenon.) This result is both reassuring in that we can usually slightly oversample and make use of low complexity algorithms, and a warning that physical limits and models cannot be abstracted away. In the conclusion we state some open research problems in the domain of sensor network cooperation.

## Chapter 2

# Background

In chapter 3, the first case study subject is localization. Localization is a very popular subject in sensor network, and with good reason. In many situation where sensor networks are deployed, we are interested in where events take place. Broadly speaking there are three approaches to locate a source: triangulation, scene analysis and proximity[15]. Triangulation requires the use of multiple sensors that provide simple data and geometry, where scene analysis makes use of more complex data streams such as images to locate items. This is the classical approach to localization. Localization via scene analysis itself is a broad research topic under computer vision. A few examples are shown in [16, 17, 18, 19]. The proximity approach was used in many sensor networks and cell phone localization problems [20]. We will focus on triangulation, where the fusing of data among sensors is more explicit and scaling of cooperation with multiple sensors is simpler.

As with any estimator (in our case a location estimator), one metric of quality is the Cramer-Rao Bound (CRB)[21]. It provides a theoretical bound of the estimator based

on the relationship between the measurement and the estimate, and the noise model of the measurement. The CRB bounds the variance of any unbiased estimator. Although we cannot compute CRB during runtime because we do not actually know the ground truth, CRB produces insight in expected system performance and verifies implementation during calibration. In higher dimensions, the inverse of the CRB may not exist when it is not full rank, but the inverse of CRB, the Fisher Information Matrix (FIM) can still be determined and inform us of the dimension where the ill condition occurred.

In sensor networks there is a new problem that previous remote sensing did not often face: sensor selection. In the classical remote sensing problem, there are far fewer sensors available. Typically we need all the sensors available in the classical case. In the sensor network setting, we may have multiple sensors detect and observe the same source. For various reasons such as reduced congestion or increased life time of the network, we want to select only a few of the sensors within the network to respond to the source. In order to decide which sensors to select in an intelligent fashion, we need to define some form of utility from each sensor. The utility obtainable from each sensor and its contribution to the overall utility depends on the application. Total utility as a function of each sensor's utility may be either sub modular or super modular [22]. Sub modular utility functions are utility functions such that the total utility from all sensors is equal to or less than the sum of the individual sensors' utility. The sensor selection problem with this kind of utility function can be solved efficiently[22]. On the other hand sensor selection under a super-modular utility function, where the aggregated utility is greater than the sum of the individual sensors' utility, is a more difficult problem. These types of utility function

arises in applications that depend on specific sequences of measurements or conditions that require a minimum number of sensors. Entropy difference[2] combines sensor selection with estimation quality by using entropy as the utility function.

Tracking is the natural next step after localization. While one way to think of tracking is a sequence of localizations, the knowledge of the previous location of the source can reduce the number of sensors we must explicitly deploy. Although we will not deal with tracking explicitly in this thesis, the sequence of past observations can be treated as other observations. However, the additional time dimension needs to be considered, both in terms of when the measurement was taken place and the time it takes to resolve an estimate of the solution. The standard approach to solving tracking problems is the Kalman Filter[23]. The Kalman Filter solves the linear state space model with Gaussian noise corrupting the measurement. Another class of newer tracking solvers is the particle filter which solves the tracking problem using the Sequential Monte Carlo Methods (SMCM) [24]. The particle filter admits a more general model compared to the Kalman Filter, with a more complex implementation. Essentially the SMCM constructs a simulation of the model, constructs the probability distribution based on the measurements and evaluates various parameters of the problem from the computed distribution at each iteration.

To analyze the consequence of distance loss, we turn to order statistics[25]. While we may place our sensors in an independent, identical distribution (iid) fashion, the distribution of the closest, the next closest, and the farthest sensors are certainly not identical. Order statistics construct the distribution of the closest to the farthest sensors from the distribution of how the sensors are scattered in the field.

If synchronism can be achieved among sensors for the sensing, the optimal combining technique is Maximal Ratio Combining(MRC)[5]. MRC effectively adds each sensor's signal to noise ratio by coherently combining the signal (e.g. add in voltage) while noise combines in power. This is the upper bound of signal combining. In some cases coherently combining is not possible due to the nature of the source or when synchronism is not available. In that case the benefit of cooperation will be reduced.

A lattice structure allows us to analyze coverage problems in arbitrarily large fields. With the rich structure in the lattice we can expand the analysis easily. We will use the hexagonal lattice due to the nice packing property[26] in the two dimensional space in our analysis. Analysis of random deployments is more challenging, because although we may place the sensors in an iid fashion, we may be subject to edge effects which makes large field analysis difficult. In the analysis of capacity of wireless networks[27] the authors use a specific deployment distribution (on the surface of a sphere) to avoid the edge effect.

In the reconstruction chapter, there are several topics in which we only made use of a very small portion of a vast subject: wavelet transform, compressed sensing and statistical techniques. In this thesis we will focus on the one dimensional version of these topics. This will simplify our problem and allow us to maintain the focus of this thesis in studying the benefit of various cooperation neighborhood sizes. The wavelet transform[28] expands upon the Fourier transform and has been an active research area. The wavelet transforms has seen extensive applications in image processing and as part of the JPEG 2000 standard [29]. The wavelet transform is very attractive in signal compression and processing, partly due to its ability to describe both high and low frequency content of the source efficiently,

compared to the Fourier transform. Many applications of compressed sensing and tutorials can be found at [30]. Compressed sensing can make use of the wavelet transform's ability to enable efficient sampling of large class of signal, although we did not implement it in that fashion in this thesis. As we will discuss in chapter 6, the success of global algorithms depends on the correctness of the model.

This idea of matching a model to the source led to the next class of algorithms: statistical algorithms. Using an iterative procedure, the algorithms can learn and approximate the underlying distribution of the parameters of interest. In many cases, however, the parameters we should use require other prior knowledge and are application specific. [31] provides an excellent introduction to such techniques.

## Chapter 3

# Uniform Contribution

### 3.1 Introduction

In a sensor network, there are multiple sensors observing the source. With multiple sensors, we may only need a subset of the sensors within the network to achieve our sensing objective. Since only a subset of the sensors is needed, the natural question becomes "which sensors should we select?" Unfortunately, as [1] points out, even when the utility function from each sensor is well behaved, the selection of the optimal subset can be challenging. Nonetheless, as we will show in this and subsequent chapters, we only need to select a few sensors to get most of the utility.

In this chapter, we will consider the case that all sensors provide identical utility, using the case study of localization to illustrate the identical utility model.

### 3.2 Uniform utility and its consequence

Suppose each sensor contributes an identical amount of utility, and the overall utility of the fused data will be the sum of the individual utilities, i.e. the utility for  $n$  sensors will be simply  $n$ .

At the  $n$ -th iteration of data fusion, the existing sensor set provides  $n - 1$  units of utility, and the relative utility  $u_r$  of the existing set to the  $n$ -th iteration is

$$u_r(n) = \frac{n - 1}{n} \quad (3.1)$$

The difference is

$$\Delta u_r(n) = \frac{2n - 1}{n(n + 1)} \quad (3.2)$$

To increase the utility  $n$  by factor  $k$ , it will require  $nk$  sensors, and the utility will approach  $nk$  in  $O(1/n)$ .

We can achieve an arbitrary utility by deploying a sufficiently large number of sensors. However, the rate of each sensor's actual contribution to the overall utility decreases geometrically as the total utility increases, as seen in figure 3.1.

The implication is that in a very dense deployment, some overlap or otherwise a reduction in individual sensor utility will not be noticeable. Figure 3.2 shows that when individual sensor utilities are uniformly distributed between 0.5 to 1.5, as the number of sensors increases, the new sensor's contribution to the overall utility still diminishes geometrically. In addition, the variation in realization of this prior utility rate due to individual sensor utility variations diminishes as the number of sensors increases due to the law of

large numbers. Figure 3.2 displays the result of 10k trials, each with 25 sensors. The bar limit is the maximum and the minimum utility for a given number of sensors out of the 10k trials.

A more in-depth example can be seen in [9]. Even in their complex deployments the saturation effect is readily seen.

There are situations where the utility from sensors increases at more than a linear rate. They arise when the number of sensors used is less than necessary to provide the desired quality of service and the underlying utility has ambiguity. A typical example is localization when the observations available are less than required to form a unique solution. Such utility functions exhibit super modular behavior [22]. Under such utility functions, the benefit of cooperation among sensors is application specific. Suppose our goal is to meet a certain fixed amount of utility, and each additional sensor provides additional utility, i.e.,  $u_{n+1} = (1 + \epsilon)u_n, \epsilon > 0$  where  $u_n$  is the utility of the  $n$ -th sensor. The overall utility  $U$  with  $N$  sensors is then simply

$$U = u_1 \sum_{i=1}^N (1 + \epsilon)^{i-1} \tag{3.3}$$

This geometric series in eqn. 3.3 is also unbounded and can also reach any arbitrary utility for sufficiently large  $N$ . Even in this case, we can see that certain sensors provide the majority of the utility, but in this case it is the last few sensors as opposed to the first few sensors.

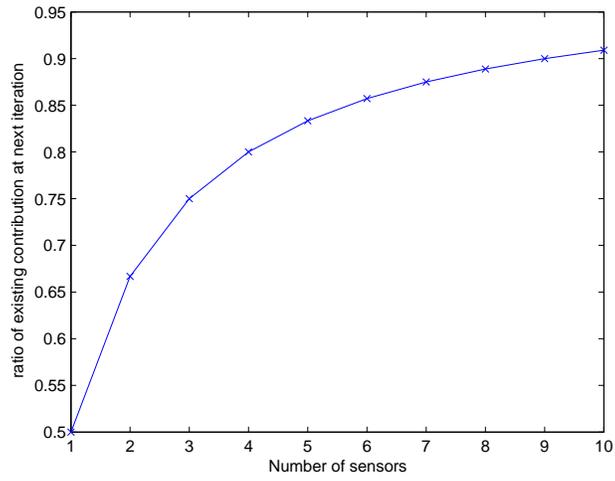


Figure 3.1: Prior utility rate to the total utility

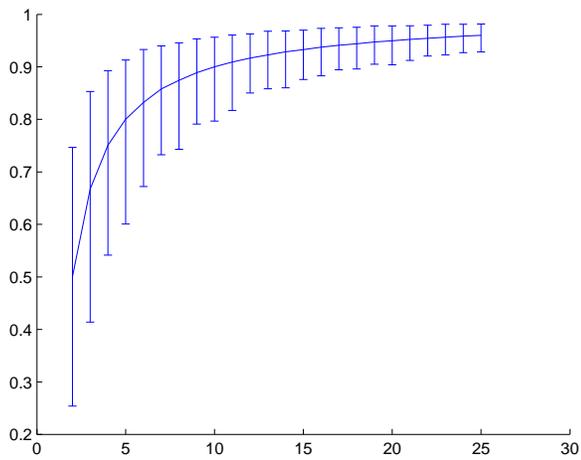


Figure 3.2: Prior utility rate to the total utility with random sensor utility

### 3.3 Localization case study

In this case study, we will consider the utility that can be derived from each sensor in a localization problem. Localization of sources can typically be achieved by three techniques: triangulation, scene analysis and proximity sensing [15]. We will focus on a subclass of the triangulation technique: range/time of travel (RNG), angle of arrival (AOA) or time difference of arrival (TDOA) measurement.

The problem of localization requires a certain number of observations be available to form a unique solution. The number of sensors required depends on the observation type.

Utility will somewhat depend on the selection algorithm. While the utility of each sensor depends on its relative location to the source, the total utility available from the sets depends on the sensor set selected. In the following section we will use selection algorithms to observe the total utility as the size of the sensor subset increases.

#### 3.3.1 Observation uncertainty model

We model the observation as a Gaussian distribution centered around the true reading. The following notation is used: the range is  $\rho_i \sim \mathcal{N}(\bar{\rho}_i, \sigma_i)$  for  $i$ -th range sensors,  $i = 1, \dots, k_R$ , the angle is  $\theta_i \sim \mathcal{N}(\bar{\theta}_i, \sigma_i)$  for  $i$ -th AOA sensors,  $i = 1, \dots, k_A$ , and the delay is denoted  $\tau_i \sim \mathcal{N}(\bar{\tau}_i, \sigma_i)$  for  $i$ -th TDOA sensors,  $i = 1, \dots, k_T$ . Each sensor may report a range, an angle of arrival or a time difference of arrival estimate. The estimation of the range or time of arrival, angle of arrival, TDOA are abstracted here, i.e., each sensor individually handles the estimation and reports the final estimated observation. We assume observation noise in each sensor is independent of that from other sensors.

By contrast, [2] did not require the noise process to be Gaussian. Noise in [2] can follow a more general distribution as long as the conditional differential entropy of observation, given the source’s location estimate, exists. [2] thus allows for a more accurate evaluation of uncertainty when the noise is non-Gaussian. Entropy difference [2] is described below for comparison as a sensor selection algorithm. With a small number of sensors, the entropy difference selection algorithm selects a set of sensors that can achieve a higher quality localization solution. When a large number of sensors cooperate to localize a source, the performance difference between selection algorithms diminishes.

[32] focused on AOA based sensors. Each of those sensors trace out two rays, originating from the sensors and angled at  $\bar{\theta}_i \pm \alpha_i/2$ ,  $\alpha_i$  the bounded uncertainty for sensor  $i$ . Using this model, the localization problem becomes a series of slices of polyhedra. The slicing of the field of view can be performed efficiently. The final localization product is a polyhedra, where its area can also be computed efficiently [33]. However, if the angle  $\alpha_i$  is too small that leads to error during selection of the slice of the polyhedra, and the algorithm will produce erroneous and inconsistent results. The inconsistency is a threat especially when there is a large number of sensors. Therefore angle  $\alpha_i$  must be selected sufficiently widely across the AOA sensor range, which may include large targets at short distance.

### 3.3.2 Localization uncertainty

Due to the observation uncertainty, localization will have limited accuracy. A common criterion for accuracy of estimating parameters is the Cramer-Rao bound. From [34],[35],[36], based on the observation uncertainty model used in 3.3.1, in a planar localiza-

tion problem, the CRB matrix and Fisher Information Matrix (FIM) for range and (AOA) localization are as follows:

$$\text{CRB} = \text{FIM}^{-1} = (\mathbf{H}^* \mathbf{H})^{-1}$$

$$\text{Range: } \mathbf{H}_{\text{RNG}} = \begin{bmatrix} \vdots \\ \frac{\mathbf{r}_s - \mathbf{r}_i}{\sigma_i \|\mathbf{r}_s - \mathbf{r}_i\|} \\ \vdots \end{bmatrix} \quad (3.4)$$

$$\text{AOA: } \mathbf{H}_{\text{AOA}} = \begin{bmatrix} \vdots \\ \frac{y_s - y_i, x_s - x_i}{\sigma_i \|\mathbf{r}_s - \mathbf{r}_i\|^2} \\ \vdots \end{bmatrix} \quad (3.5)$$

where  $\mathbf{r}_s = [x_s, y_s]$  is source location and  $\mathbf{r}_i = [x_i, y_i]$  is  $i$ -th sensor's location.  $\sigma_i^2$  is the  $i$ -th sensors observation variance.

With time difference of arrival (TDOA) localization, in the case of unknown propagation velocity, the CRB matrix for location parameter and propagation velocity can be expressed as follows [37]:

$$\text{TDOA: } \mathbf{H} = \frac{1}{v} \begin{bmatrix} \vdots \\ \frac{1}{\sigma_i} \left( \frac{\mathbf{r}_s - \mathbf{r}_i}{\|\mathbf{r}_s - \mathbf{r}_i\|} - \frac{\mathbf{r}_s - \mathbf{r}_{\text{ref}}}{\|\mathbf{r}_s - \mathbf{r}_{\text{ref}}\|} \right), -\tau_i \\ \vdots \end{bmatrix} \quad (3.6)$$

where  $\mathbf{r}_{\text{ref}} = [x_{\text{ref}}, y_{\text{ref}}]$  is the reference sensor location,  $v$  is the propagation velocity and  $\tau_{i\text{ref}}$  is the time difference between the  $i$ -th sensor and the reference sensor.

Note that while the CRB for range and AOA sensor modes can be computed without actual observations, it does require actual source location in all sensing modes. Hence we cannot compute the true CRB in any actual implementation. Nevertheless, the CRB can be used to compare the performance of sensor selection algorithms, by comparing each selections' best possible performance with an ideal estimator in term of localization co-variance. In contrast, [2] uses the probability distribution of source location as the metric of localization uncertainty while [32] uses the area of the source's possible location as the metric. Unlike true CRB, both entropy difference and source location uncertainty area under the bounded uncertainty model can be computed in an actual implementation if distributions of sensors' observations are accurate or the uncertainty in observation from sensors is indeed bounded.

Since  $\text{FIM}_{\text{RNG}}$  and  $\text{FIM}_{\text{AOA}}$  have the same dimension, for mixed mode operation with range and AOA sensors, we can simply add the FIM from each mode. If we also have TDOA sensors, we can zero pad FIM from the range or AOA mode with a row of 0's and a column of 0's, i.e.

$$\mathring{\text{FIM}}_{\{\text{RNG}, \text{AOA}\}} = \begin{bmatrix} \text{FIM}_{\{\text{RNG}, \text{AOA}\}} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \quad (3.7)$$

$$\text{FIM} = \mathring{\text{FIM}}_{\text{RNG}} + \mathring{\text{FIM}}_{\text{AOA}} + \text{FIM}_{\text{TDOA}}$$

Note that the above derivation did *not* explicitly assume any distance loss in signal quality, i.e.  $\sigma_i$  in reality may also be a function of distance  $\mathbf{r}$ . This simplification has little impact when the source is 'far' away from the sensors group and most sensors actually

observe similar signal strength.

However this simplification has considerable impact when the relative distances between the source and the sensors vary significantly among sensors, which will be the case when the source is ‘near’ the sensor group. This distance dependent interaction is affected by the distribution of sensors around the source and will be considered in section 4.1.

### 3.3.3 Localization Utility Simulation

In the following simulation, the source was placed within the field of sensors. We ran 300 trials of the experiment. In each trial, there were 16 RNG and 16 AOA sensors, with .08 standard deviation on observations for both types of sensors. Therefore the  $d_{be}$  is 1. AOA sensors within radius 1 will be selected first, followed by the entire set of RNG sensors followed by the remaining AOA sensors. The sensors are distributed uniformly over a  $[-1, 1] \times [-1, 1]$  square and the source is placed uniformly over a  $[-.1, .1] \times [-.1, .1]$  box.

$\lambda$  in Fig. 3.3 is the sum of the eigenvalues of the FIM and is the utility metric for this localization simulation. Note the rapid saturation of the utility after a few sensors, regardless of the selection algorithm used in selecting sensors. In this case the minimum number of sensors is three for the RNG sensors.

Several algorithms were used to select sensors in a sequential fashion among the entire set of sensors. The different selection algorithms show that the underlying localization problem renders the sensor selection problem trivial when sensors are sufficiently densely deployed. The algorithms we used are as follows:

## Random

We simply pick sensors randomly. This is the simplest method; it requires no prior information. The density of the deployment determines the success of this method. In particular, this method will be successful in a dense deployment, and will fail easily in a sparse deployment.

## Entropy Difference

From the observation model, each sensors' observation has a certain probability distribution. From [2], we may use a heuristic based on information theory to sort sensors according to their potential benefit in improving our accuracy in the localization problem. This method considers the problem in its entirety; both the sensor's observation variance and the geometric factors are considered.

This method requires two parts. First the entire observable space has to be discretized once to compute  $\mathbf{H}_i^v$  ((9) in [2]) for each sensor to compute the a priori entropy of observation.

$$H_i^v = - \int p(z) \log p(z) dz$$

where  $z$  is the field of view of the sensors. This  $\mathbf{H}_i^v$  only depends on sensors location and the geometry of the observable space.

At each iteration, based on the previously picked sensor, across the entire observable space, we need to compute  $\mathbf{H}_i^s$  ((11) in [2]), where  $\mathbf{H}_i^s$  represents the entropy of the sensor observation given that the source location is estimated based on knowledge available

up to present:

$$H_i^s = - \int p(z|\hat{x}) \log p(z|\hat{x}) dz$$

where  $\hat{x}$  is the latest maximum likelihood estimate of source location.

[2] detailed how  $H_i^v - H_i^s$  approximates the mutual information comparison at each sensor and the difference from the actual mutual information. All of this is very computing intensive. In fact, according to [2], it's  $O(w^3)$ , assuming the observable space is gridded into a  $n \times w$  matrix. If sensors are to compute the entropy difference in a distributed fashion, each sensor requires the probability distribution of the source location. At the end of each iteration the probability distribution of the source location will be updated with the selected sensor's observation.

### **Nearest Sensor First**

Another heuristic method is to sort the sensors according to their distance from the estimated source location, then pick the closest sensor first. Presenting the selection algorithm as an optimization problem, we want to select the  $i$ -th sensors that

$$\min_i \|\mathbf{r}_s - \mathbf{r}_i\|$$

All that is required is some kind of estimation of source location and all sensors locations. The complexity of this is  $O(m \times p)$ , where  $m$  is the number of sensors and  $p$  is the dimension of the localization space if we execute all the distance calculations only one time and do not update the source location as we progress. Of course, in practice to guard

against large initial errors some additional updates are advisable for robust operation.

### **Angle-Range-Angle**

Yet another heuristic method is to select some AOA sensors first, then select all the RNG sensors and then the remaining AOA sensors. Suppose for all RNG sensors the observation variance is  $\sigma_{\mathbf{R}}$  and for all AOA sensors the observation variance is  $\sigma_{\mathbf{A}}$ . All RNG sensors are equivalent in terms of their utility toward the localization application, since  $\mathbf{h}_{\text{RNG}}$  are unit vector scaled by  $\sigma_{\mathbf{R}}$ . On the other hand, AOA sensors that are closer to the source provide more utility than the farther away counterparts.

### **3.3.4 Localization Conclusion**

A typical realization of utility progress, with a variety of sensor selection algorithms is shown in Fig. 3.3. Fig. 3.4 shows the number of sensors needed for a variety of sensor selection algorithms to achieve 90% of the utility achieved by using all sensors.

The selection algorithm can have a great impact on the overall aggregated performance because the first few sensors are going to indicate the aggregated performance. In this stage of selection, a more expensive algorithm is worthwhile because we can reach the desired quality of service faster. We don't need the best at each stage, but a greedy algorithm probably should work. In our localization example, entropy difference performed best via a more accurate prediction of sensors contribution given the selected sensors set. While it is more complex to perform at each iteration, entropy difference meets the QoS with only half as many sensors as other algorithms. On the other hand, if we will use a large number of sensors, the particular selection algorithm has little impact because utility

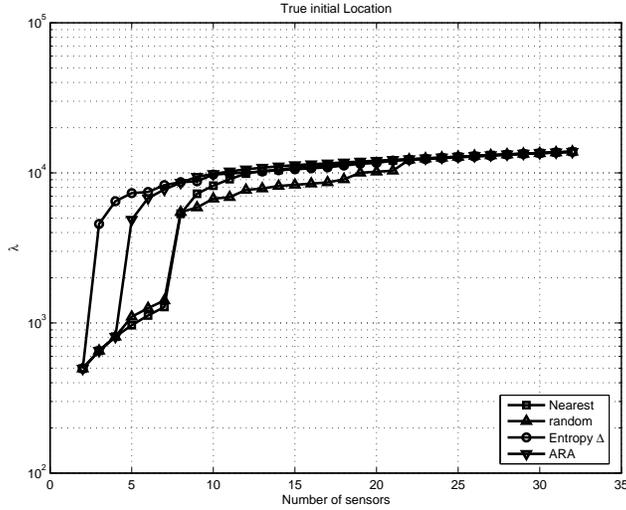


Figure 3.3: One realization of of localization utility

grows at a slow rate after the first few sensors.

Clearly the first few sensors contributed the majority of the utility of the data fusion, under this simplified utility function. If some sensors actually have a higher utility function than other sensors, even fewer sensors will contribute most of the utility. The selection algorithm is important for selecting those first few sensors. In the above example, the entropy difference selection method performed the best, ARA is next and select the next nearest sensor performs similarly to randomly selecting sensors. Note that the nearest sensor performs so poorly partly due to the fact that there is no distance dependency for the utility of the sensor observations. However, even with a simple selection algorithm, a few additional sensors can achieve the desired utility. Consequently, while there are some differences among the algorithms' performance with a small number of sensors, the utility function here penalizes the simple selection algorithm only slightly compared to more complex algorithms, assuming a few more sensors are available.

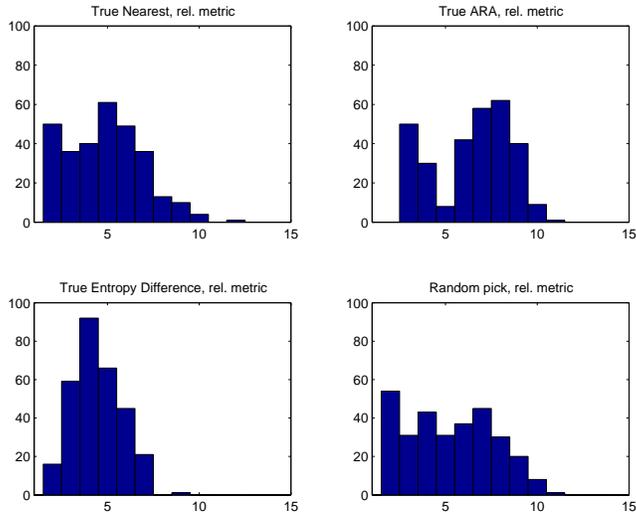


Figure 3.4: Number of sensors to achieve 90% of total utility

From the above exercise, we can see that a few sensors in addition to the minimum required to resolve the source location uniquely will be sufficient to localize the source, echoing a result observed in [32]. In the next chapter, we will consider cooperation utility with distance dependency.

## Chapter 4

# Non-Uniform Utility Metric

### 4.1 Introduction

In this chapter, we will consider non-uniform sensor utility that arises due to the sensors' geometric distribution and the resulting distance loss effect on the expected utility. Order statistics will play a key role to transform the sensors distribution to the expected utility.

In this scenario, we place the source at the origin, and sensors are placed in an IID fashion. We will consider two distributions: uniform disk and two dimensional Gaussian distribution. We then compute the expected  $k$ -th closest sensor with order statistics. That in turn allows us to compute the distribution of the utility of the  $k$ -th sensor.

From this expected utility we then show the rapid fall off of sensor utility from the nearest sensor to the farther away sensor. This observation implies that the closest few sensors contribute significantly larger utility than those that are farther away.

Additionally, we will apply fading to the observations. Since fading can degrade or

improve signal strength significantly, we will see that cooperation of the few closest sensors is necessary because the closest sensor may experience fading that will erode the distance advantage a sensor may otherwise enjoy.

From the order statistics results, we can also compute the expected total utility. We observe that large scale cooperation utility is always bounded when the distance loss exponent is greater than 2 for fixed density.

In addition, we will observe that a small number of sensors cooperating helps in overcoming outage due to fading.

## 4.2 Sensor Utility Statistics

### 4.2.1 Order Statistics

The distribution of the  $k$ -th statistic out of  $n$  IID drawn random variables can be written as follows [25]:

$$f_{X_{(k)}}(x) = n \binom{n-1}{k-1} F(x)^{k-1} (1-F(x))^{n-k} f(x) \quad (4.1)$$

From eqn. 4.1, we can obtain the distribution for the nearest  $k$ -th sensor distance to the source given the number of sensors that are drawn, along with the sensor-source distance distribution, assuming all the sensors are drawn in an IID fashion. In the following we will consider two distributions: sensors are distributed in the unit disk uniformly, and sensors are distributed according to a normal distribution in a plane.

### 4.2.2 Uniform Disk

#### Layout and Assumptions

We will assume a source is at the origin and all sensors are distributed uniformly within the unit disk. The utility function for a given sensor is the distance of the sensor and the source, i.e.  $r_i^{-\alpha}$  for the  $i$ -th sensor which is  $r_i$  away from the source and  $2 \leq \alpha \leq 4$ .

#### Theoretical distribution of distance

All the sensors are placed in an IID manner, and the distances are distributed according to  $f_R(r) = 2r, r \in [0, 1]$ , a triangular distribution. The ordered statistic for the  $k$ -th closest sensor distance for  $n$  sensors total is as follows after putting the appropriate terms into eqn. 4.1.

$$f_{R(k)}(r) = n \binom{n-1}{k-1} r^{2(k-1)} (1-r^2)^{n-k} 2r \quad (4.2)$$

As fig. 4.1 shows, the first few distributions are similar and eqn. 4.2 matched well with the simulation. This similarity between the first few distributions is the key to understanding how to overcome fading as will be discussed in section 4.3.

### 4.2.3 2D Gaussian

#### Layout and Assumptions

We continue to assume a source is at the origin but now all the sensors are distributed in a planar standard normal distribution. The utility function for a given sensor is

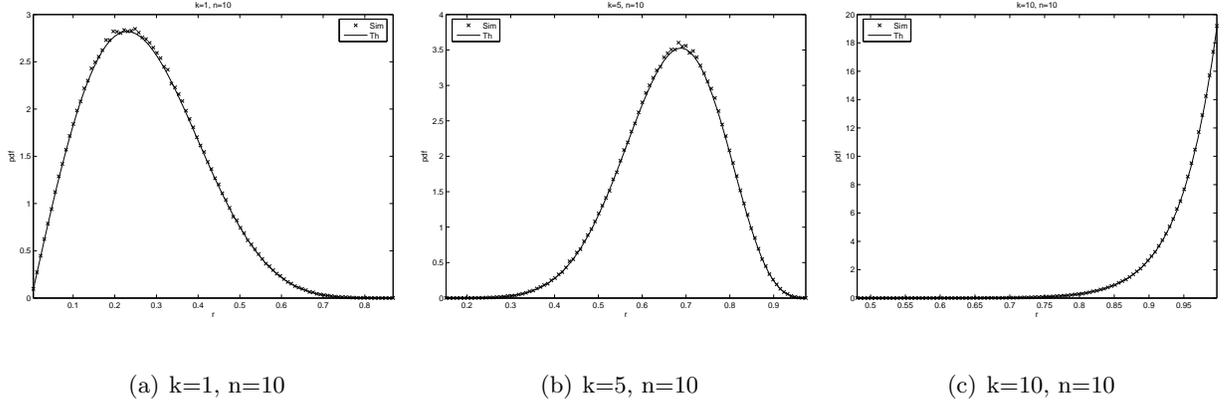


Figure 4.1: Simulation and theoretical order statistic of distance to origin, uniform disk the distance of the sensor and the source, i.e.  $r_i^{-\alpha}$  for the  $i$ -th sensor which is  $r_i$  away from the source and  $2 \leq \alpha \leq 4$ .

### Theoretical distribution of distance

All the sensors are placed in an IID manner, and the distance is distributed according to a Rayleigh distribution, with  $f_R(r) = re^{-r^2/2}$ . The extreme ordered statistics are as follows.

$$f_{R_{(k)}}(r) = n \binom{n-1}{k-1} (1 - e^{-r^2/2})^{k-1} (e^{-r^2/2})^{n-k} r e^{-r^2/2} \quad (4.3)$$

As fig. 4.2 shows, the first few distributions are also similar to each other. This distribution also shares a similar shape with the uniform disk distribution, with the exception that we no longer have a hard boundary limitation as in the disk model. However, our interest is in those that are close to origin. Thus the tail of the distribution has little impact.

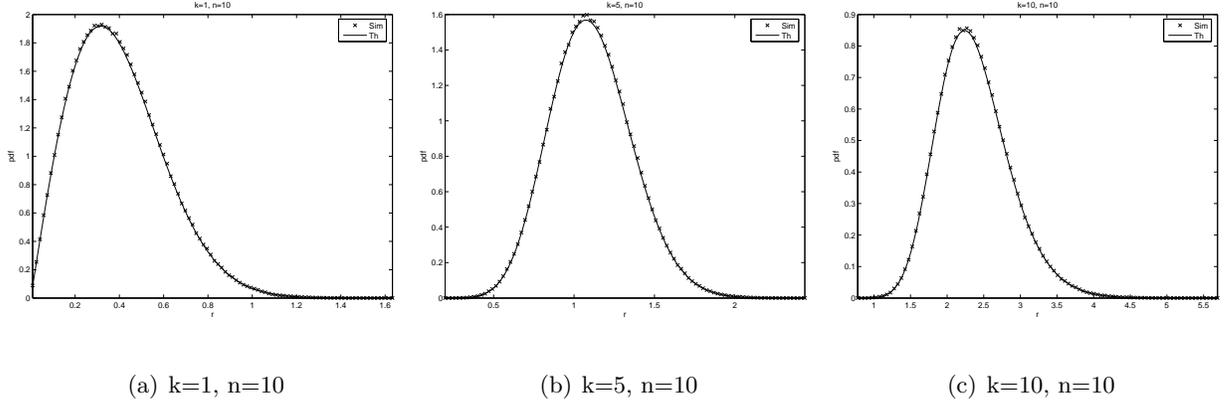


Figure 4.2: Simulation and theoretical order statistic of distance to origin, planar normal

#### 4.2.4 Expected Utility by varying k, n

The expected utility of the  $k$ -th sensor is

$$E[u_k] = \int r^{-\alpha} f_{R(k)}(r) dr \quad (4.4)$$

From eqn. 4.4, and the respective order statistic distributions from eqn. 4.2 and 4.3, we obtain the following figures, illustrating evolution of utility derived from the nearest sensor as the total number of sensors increases in the respective environments.

As shown in figures 4.3 and 4.4, both distributions behave similarly. Both experience a sharp increase in utility initially, and then the relative utility growth diminishes.

As seen from the figures 4.5 and 4.6, the utility is dominated by the nearest sensor. The relative utility is plotted, where the nearest sensor is the reference, and the level curve is the utility *below*, the reference, in dB.

The implication of the above result is that the one or two sensors that are closest to the source will generate most of the utility. This further implies that cooperation will not be an effective means to increase the utility of sensors. Thus cooperating beyond necessary,

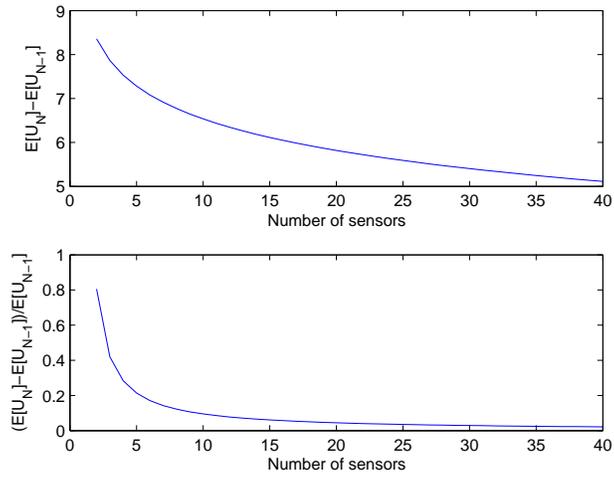


Figure 4.3: Nearest sensor expected utility evolution from  $n$  to  $n+1$  in a disk

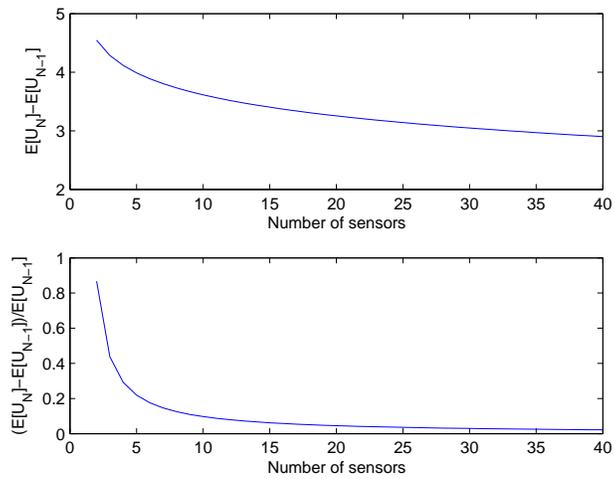


Figure 4.4: Nearest sensor expected utility evolution from  $n$  to  $n+1$  in a 2D normal distribution

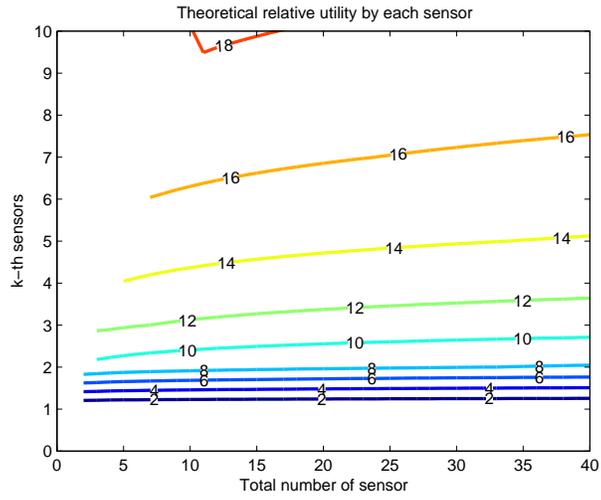


Figure 4.5: Relative utility in disk

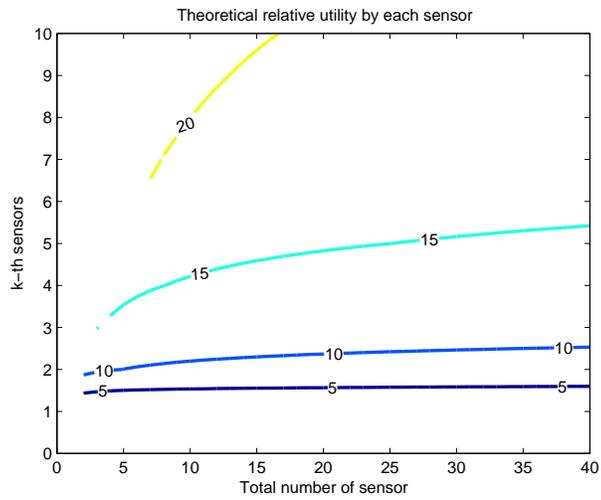


Figure 4.6: Relative utility in 2D normal distribution

e.g. the minimum number of sensors required to uniquely localize a target, will not provide much increase in utility. However, some level of cooperation should be considered in a sensor network to defend against uncertain environments as discussed below.

### 4.3 Cooperation

Assume that all sensors have equal performance and the only differences among the sensors are their distances to the source (at origin). Thus the signal to noise ratio differs from one another only by their distance loss, i.e. the SNR for the  $k$ -th sensor is  $\text{SNR}_k = S_0/N_0 \times r_k^{-\alpha}$ ,  $2 \leq \alpha \leq 4$  and  $r_k$  is the distance between the source and sensor  $k$ . Using maximal ratio combining (MRC), which is the optimal combining method, the fusion solution yields the sum of the SNR, i.e.

$$\text{Total SNR} = \sum_{k=1}^n \text{SNR}_k = S_0/N_0 \sum_{k=1}^n r_k^{-\alpha} \quad (4.5)$$

With this observation, the evolution of  $\sum_{k=1}^N r_k^{-\alpha}$  can yield insight on how the total utility changes as the number of sensors increases.

Note that this is not the same as equal gain combining. Equal gain combining, where all the sensors are weighted equally, will yield a lower SNR than the MRC.

One infelicitous environmental factor is fading. Fading can occasionally cause significant degradation to signal strength. Here we consider the utility function ( $r^{-\alpha}$ ) is multiplied by a fading factor  $g$  distributed according to the Rayleigh distribution:

$$f_G(g) = g/\sigma_f^2 \exp(-g^2/(2\sigma_f^2))$$

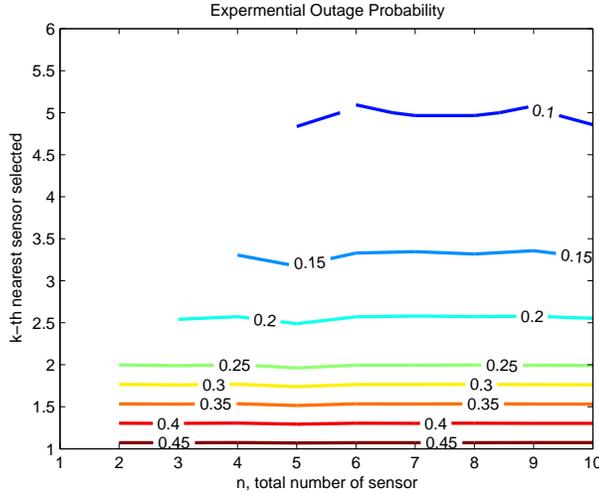


Figure 4.7: Fading outage in a disk

For a given geometry, we will draw a set of Rayleigh distributed random variables to simulate the fading effect and collect the statistics over multiple instances of the fading. We declare an outage if the sum of the faded utility is less than the nearest utility when there is no fading.

In figures 4.7 and 4.8, outage probability, as defined above, is shown with  $\sigma_f = 0.8$ . Both types of distributions experience improvement with a small number of collaborators and differ only in the tail region when outage is below 15%. As the number of sensors needed to mitigate fading increases, the difference between the two order statistics become apparent.

Not surprisingly, the outage is independent of the number of sensors in the entire deployment, as seen in figures 4.7 and 4.8 that the given probability of outage depends only on number of sensors used ( $k$ ), and not on the total number of sensors ( $n$ ). That is due to the outage definition above, where outage is related to the nearest sensor utility in

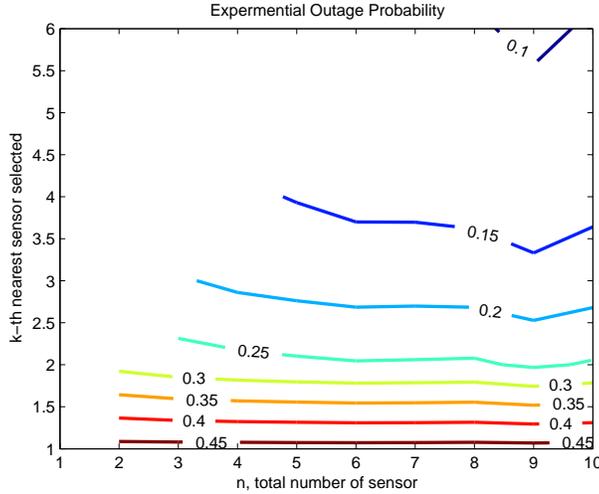


Figure 4.8: Fading outage in 2D normal distribution

a non-fading environment. For a given fading environment, a few sensors cooperating is necessary to provide acceptable performance. This is in contrast to the previous scenario. In the scenario where there is no fading, the nearest sensor alone is sufficient. Nonetheless, even in this case a small number of sensors suffices.

Suppose we define outage as a certain quality of service (QoS), in this case as the expected utility from two sensors in a non-fading environment. The outage is shown in figures 4.9 and 4.10. With the fixed QoS, it is not surprising that as the total number of sensors or the number of sensors used in cooperation increases the outage decreases. Note also that as the total number of sensors increases, the number of sensors for actual cooperation can be reduced in order to reach the target QoS. That is achieved by sensors being closer to the source such that those sensors can provide the target QoS even in a fading environment.

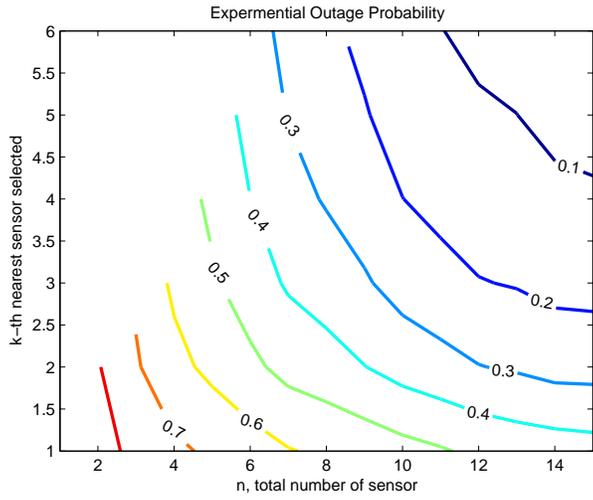


Figure 4.9: Fading outage in a disk, fixed goal

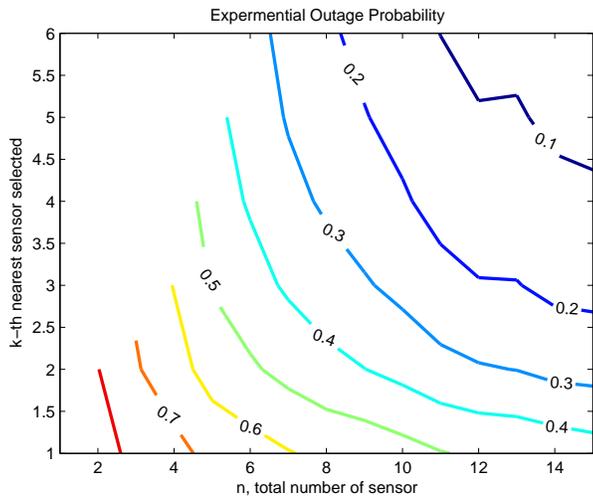


Figure 4.10: Fading outage in 2D normal distribution, fixed goal

## 4.4 Very large number of sensors

Continuing with utilities from sensors that depend on distance loss, in the following we show that the utility increases logarithmically as the total number of sensors increases when the density of deployment remains constant.

When we consider total utility, we can dispense with order statistics considerations, because the aggregated result averages across the entire set of sensors.

It can also be shown by noting that the combinatoric term in the order statistics, when we sum from sensor 1 to  $n$ , drops to 1, in particular,

$$\sum_{k=1}^n \binom{n-1}{k-1} F_R(r)^{k-1} (1 - F_R(r))^{n-k} = 1 \quad (4.6)$$

by the binomial theorem. Therefore the aggregated total utility from all  $n$  sensors can simply be obtained by multiplying the expected utility of one sensor by  $n$ .

Note that when considering utility functions in the form of  $r^{-\alpha}$ , the rate that the probability approaches 0 when distance approaches 0 must be taken into account. A uniform disk, where the distance distribution is a triangular distribution, does not decrease to 0 fast enough, thus a minimum distance  $\epsilon$  is introduced to avoid unbounded utility when taking the limit on the number of sensors.

For deployment within a uniform annulus with radii  $\epsilon$  and  $R$ , the probability distribution function is

$$f_R(r) = \frac{2(r - \epsilon)}{(R - \epsilon)^2} \quad (4.7)$$

A sensor's expected utility in such a deployment, for  $\alpha > 2$  is simply

$$\begin{aligned} \mathbb{E}[u] &= \int_{\epsilon}^R r^{-\alpha} f_R(r) dr \\ &= \frac{2}{(R-\epsilon)^2} \left[ \frac{r^{2-\alpha}}{2-\alpha} - \frac{\epsilon r^{1-\alpha}}{1-\alpha} \right]_{\epsilon}^R \end{aligned} \quad (4.8)$$

and for  $\alpha = 2$ , the expected utility is

$$\mathbb{E}[u] = \frac{2}{(R-\epsilon)^2} \left[ \log \frac{R}{\epsilon} + \epsilon(R^{-1} - \epsilon^{-1}) \right] \quad (4.9)$$

In the case of constant density  $M = n/(\pi R^2 - \pi \epsilon^2)$ , the total utility is  $n$  times the expected utility of one sensor (eqn. 4.8, 4.9),

$$\begin{aligned} &\lim_{R \rightarrow \infty} \frac{2n}{(R-\epsilon)^2} \left[ \frac{r^{2-\alpha}}{2-\alpha} - \frac{\epsilon r^{1-\alpha}}{1-\alpha} \right]_{\epsilon}^R, \alpha > 2 \\ &= \frac{2M\pi\epsilon^{2-\alpha}}{(\alpha-2)(\alpha-1)} \end{aligned} \quad (4.10)$$

$$\begin{aligned} &\lim_{R \rightarrow \infty} \frac{2n}{(R-\epsilon)^2} \left[ \log \frac{R}{\epsilon} + \epsilon(R^{-1} - \epsilon^{-1}) \right], \alpha = 2 \\ &= \lim_{R \rightarrow \infty} \frac{2M\pi(R^2 - \epsilon^2)}{(R-\epsilon)^2} \left[ \log \frac{R}{\epsilon} - 1 \right] \\ &= \lim_{R \rightarrow \infty} 2M\pi \left[ \log \frac{R}{\epsilon} - 1 \right] \end{aligned} \quad (4.11)$$

The above limits in eqn. 4.10 show that for  $\alpha > 2$  the total utility is bounded regardless of the number of sensors in a fixed density disk. For  $\alpha = 2$  the total utility is unbounded, and grows logarithmically with respect to the number of sensors in a fixed density disk.

Fig. 4.11 shows the aggregated utility of  $n$  sensors for  $\alpha = 2$ . As the number of sensors increases, the maximum distance  $R$  increases to maintain constant density. The linear behavior in the large number of sensors regime confirms that the utility grows logarithmically.

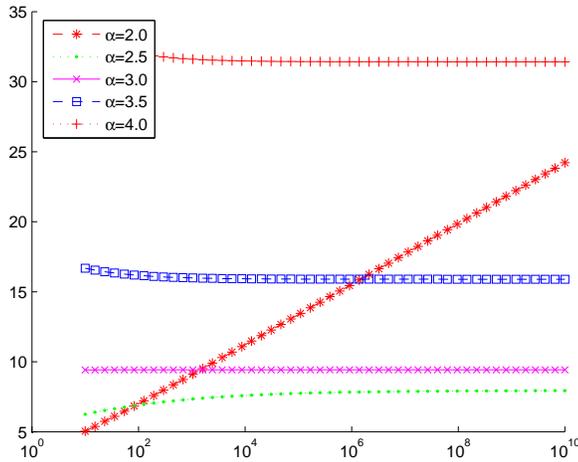


Figure 4.11: Expected aggregated utility from constant density in an annulus

Also note that the utility appears to *decrease* slightly as the number of sensors increases for larger  $\alpha$  values. Intuitively, for a larger  $\alpha$  value, where distance loss is more significant, the optimal sensor is more critical to the overall utility. As the number of sensors increases, the expected distance for the nearest sensor *increases* slightly according to the order statistic due to the exclusion zone  $\epsilon$  in the distribution, shown in Fig. 4.12. In reality, utility would be flat.

## 4.5 Non-uniform utility conclusion

In summary we have shown that cooperation utility is bounded in most cases for a given distance. This limitation is based on distance loss. Therefore large scale cooperation is both unnecessary and ineffective. However, a local scale cooperation can provide great benefit to deal with degradations such as fading. In the next chapter will we see how small scale cooperation enhances coverage area.

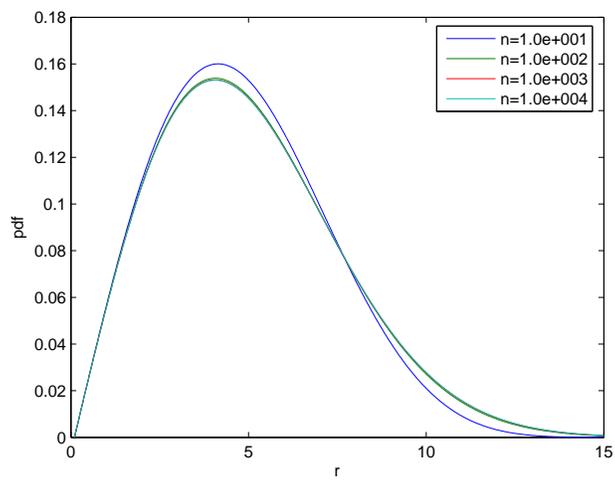


Figure 4.12: PDF of the nearest sensor distance

# Chapter 5

## Coverage

### 5.1 Introduction

The coverage problem is a mixture of both uniform and non-uniform utility. On the one hand, if we assume total coverage area as the utility metric and each sensor provides identical coverage area, then this is a uniform utility scenario. At this scale the ‘cooperation’ among sensors is by merely sensing a different area. On the other hand, if a more advanced cooperation method is available, then sensors coverage utility is non-uniform, and is modified by how sensors cooperate with their neighbors. In this chapter we will focus on the latter setup: how does local cooperation among neighbors impact coverage provided.

We will consider a regular lattice deployment and a random deployment of sensors to observe the effect of cooperation on coverage area. As we have seen in previous chapters, small scale cooperation among a handful of sensors can provide a noticeable increase in utility. On the other hand, large numbers of sensors cooperating to sense a common source provides diminishing returns. In both deployment scenarios we will observe that a small

scale cooperation will provide enhanced coverage area to the field.

## 5.2 Prior art on coverage

The coverage problem had been studied extensively in the literature in various forms. One typical question is the density necessary for a certain coverage rate [7, 8, 9]. Each sensor operates independently within its coverage area (typically a disk). In [38, 39] the impact on the overall coverage by a more realistic coverage model from each sensor was explored. A heterogeneous mixture of sensor capability and their impact on the network coverage is considered in [40].

Localization requires multiple sensors to achieve a solution. [10] considered the coverage problem in the localization context. Similarly, for reliability purposes, it is desirable to have redundant sensors so that the network can continue to achieve its objective even when some sensors become unavailable. [11, 12] explored this particular type of problem. Typically this type of problem is cast as a  $k$ -coverage problem. In contrast to an independent sensors setup, in this problem we requires  $k$  sensors providing identical utility, whether for the communication link or sensing coverage.

Another large set of literature on coverage is on controlling placement of sensors to achieve such coverage in both static placement [41] and dynamic sensor networks with robots [42, 43, 44]. Another deployment approach is to deploy sensors in a random fashion [45].

In the following, the effect of cooperation among sensors to improve coverage is considered. Similarly to [45], we will consider using multiple sensors cooperating to

achieve an increased coverage area. In particular, we will use multiple sensors to provide an equivalent one sensor coverage. This objective is in between the independent sensor coverage and the  $k$  coverage problem. In the independent sensor approach, the source level within the coverage area must reach the detection threshold at a given sensor. Likewise in the  $k$  coverage problem,  $k$  sensors observe the source above the detection threshold within the coverage area. In contrast, we will consider the case where we are allowing several sensors cooperating where *none* of the sensors observe the source at the detection threshold. We will declare a source within the coverage area if after combining the cooperating sensors measurements, the combined signal meets the same detection threshold.

### 5.3 Sensor model and utility metric

We will model the coverage of a sensor as a disk of radius  $r$  and a distance loss exponent of  $\alpha$  throughout this chapter. For cooperation in sensing among sensors, we will assume maximal ratio combining, the same cooperation model as section 4.4. In particular, cooperation SNR will be the same as equation 4.6.

We will define the utility function as follows: for a given point  $\mathbf{x}$ , with  $m$  cooperating sensors, then point  $\mathbf{x}$  is covered if the combined SNR from the  $m$  highest SNR sensors is equivalent to being  $r$  away from one single sensor. We will further assume the difference between sensors' SNR is due only to the distance of the sensors. Therefore the  $m$  closest sensors will provide the highest SNR. Let  $\mathbf{s}_i$  be the location of the  $i$ -th sensor. Then the point  $\mathbf{x}$  is covered under  $m$  cooperation when,

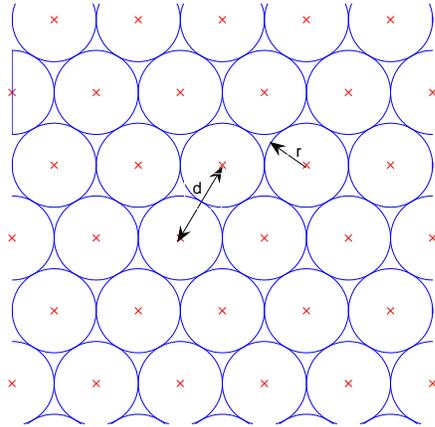


Figure 5.1: Sensor in Lattice example

$$\sum_{i=1}^m r_i^{-\alpha} \leq r^{-\alpha}, r_i = \|\mathbf{s}_i - \mathbf{x}\|_2 \quad (5.1)$$

## 5.4 Regular deployment

We place the sensors according to a hexagonal lattice. Hexagonal packing of the disk places all neighboring sensors at equal distance, greatly simplifying the analysis.

In particular, each sensor  $s$  will have six neighbors, distributed evenly at a distance of  $d = kr$  away from  $s$ . The neighbors are separated by  $\pi/3$  radians angularly from each other with respect to the center sensor.  $k$  scales the spacing between neighbors and the coverage area. Fig. 5.1 shows an example of the lattice, with  $k = 2$ . We will consider the lattice of fixed density and use  $k$  as the utility metric in this section. The scaling factor  $k$  is proportional to the coverage area.

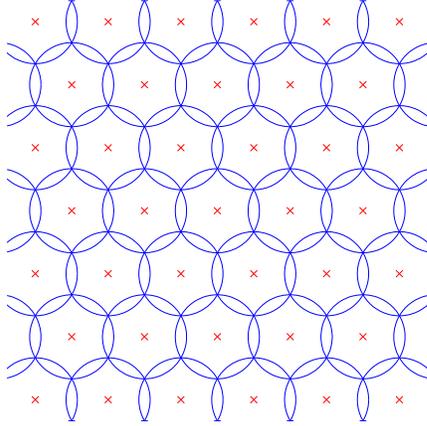


Figure 5.2: Complete coverage with independent sensors

#### 5.4.1 Independent sensors

In fig. 5.1, we will notice there is some space that is at a distance greater than  $r$ . The distance between the farthest point away from any sensor to the center of a sensor is

$$d\sqrt{3}/3 \tag{5.2}$$

by the lattice configuration. In order to place *all* points within a sensor range  $r$ ,  $k$  must be less than or equal to  $\sqrt{3}$ . Therefore  $k = \sqrt{3}$  is the lower bound of the coverage area with no cooperation and no void in coverage. Fig. 5.2 shows the lattice and coverage with  $k = \sqrt{3}$ .

The overlap area of per each sensor with this complete coverage is  $2r^2(\pi - 3\sqrt{3}/2)$  per sensor. Compared to the individual sensor area of  $\pi r^2$  for each sensor there is approximately 35% of resources that provide no added utility because those areas are also covered by other sensors.

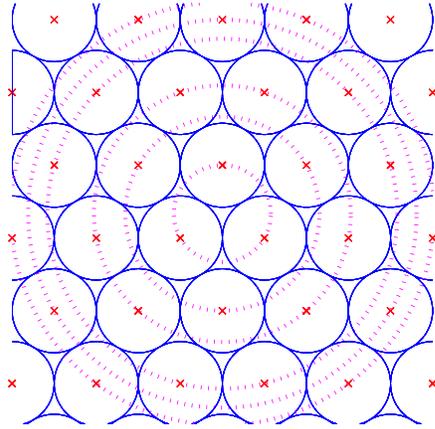


Figure 5.3: Sensors surround origin

#### 5.4.2 Local cooperation sensors

By enabling cooperation among sensors we can set  $k \leq \sqrt{3}$  and still achieve complete coverage. Without loss of generality, we can select one of those points that is farthest from the nearest sensors and place it at the origin.

Let's consider a cooperation size of 3 sensors. With three sensors cooperating,  $k$  can be set as follows

$$3(kr\sqrt{3}/3)^{-\alpha} = r^{-\alpha}$$

$$k = (3)^{1/\alpha}\sqrt{3}$$

This will maintain maintain complete coverage. In addition to eliminating overlap of coverage areas by sensors, the single sensor's coverage areas actually do not contact their neighbors.

If we add additional sensors for cooperation, those additional sensors will be farther away. For a given ring, there will be more of such sensors. The next closest set of sensors

Table 5.1: Number of sensors and the ring distance multiplier

Ring	$n_c$	$d_n$
1	3	1
2	3	4
3	6	7
4	6	13
5	3	16
6	6	19

are  $d\sqrt{4/3}$  away from the origin. At 6 sensors cooperation,  $k$  can be set up to

$$\begin{aligned}
 3(kr\sqrt{3}/3)^{-\alpha} + 3(kr\sqrt{4/3})^{-\alpha} &= r^{-\alpha} \\
 k &= \sqrt{3} \left[ 3 + \frac{3}{2^\alpha} \right]^{1/\alpha}
 \end{aligned}$$

Continuing to the next few rings, the distance  $d\sqrt{d_n/3}$  and the number of sensors  $n_c$  on that ring are listed in table 5.1. The maximum scaling factor that achieves complete coverage for the first few rings is

$$k = \sqrt{3} \left[ \sum_{\text{ring}} n_c (\sqrt{d_n})^{-\alpha} \right]^{1/\alpha} \quad (5.3)$$

The scaling factor  $k$  suggests the coverage area that can be achieved at a given level of cooperation. In fig. 5.4, the  $k$  that resulted from various numbers of sensors cooperating and different distance loss exponents were plotted. The diminished return with respect to the number of sensors is apparent.

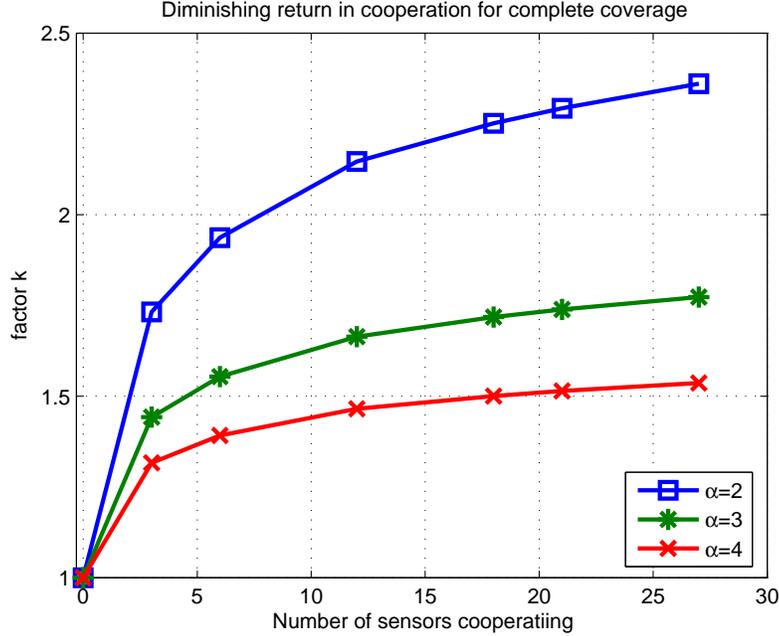


Figure 5.4: Diminishing return of cooperation

### 5.4.3 Large scale of cooperation between sensors

We can express the cooperation and scaling trade off as

$$\sum m_n d_n^{-\alpha} = r^{-\alpha} \quad (5.4)$$

where  $d_n$  is the distance of the  $n$ -th ring of sensors. From the setup in the previous section, we will use scaling factor  $k$  as the metric of cooperation which is identical for the entire field. The differences among each ring of sensors are due to their distances from the origin, expressed as  $g_n$ . Although each ring of sensors actually lies on a hexagon, we will use the closest sensor as the distance to that ring to bound the utility provided by the  $n$ -th ring. Substituting for  $d_n$  by  $kg_n r$ , eqn. 5.4 reduces to

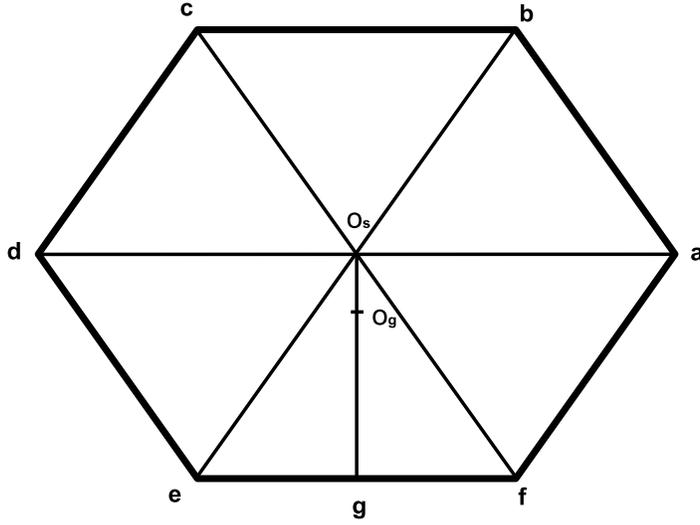


Figure 5.5: The closest point of the hexagon to the origin

$$\begin{aligned}
 k^{-\alpha} \sum_n m_n g_n^{-\alpha} &= 1 \\
 k &= (\sum_n m_n g_n^{-\alpha})^{1/\alpha}
 \end{aligned}
 \tag{5.5}$$

### Computing $g_n$

The entire sensor hexagonal lattice can be described as rings of sensors with a sensor at the origin. At each ring  $n$ , there are  $6n$  sensors with  $n \leq 1$ . Sensors at the hexagon vertices will be  $nd$  away from the origin. Distance of the sensors to the origin sensor at the  $nd$  ring can be computed by scaling a normalized hexagon. We will use the hexagon shown in figure 5.5 to evaluate the distance from the sensors to the origin.

Define  $j$  as a sensor in the  $n$ -th ring,  $j = 0, \dots, (6n - 1)$ . Segment  $s$  as  $\lfloor j/n \rfloor$  where  $\lfloor \bullet \rfloor$  is the greatest integer smaller or equal to  $\bullet$ . The segments corresponding to the edge

Table 5.2: Sensor Location in lattice

$s$	$\mathbf{V}$		$\mathbf{m}$	
0	1	0	-1	$\sqrt{3}/2$
1	1/2	$\sqrt{3}/2$	-1	0
2	-1/2	$\sqrt{3}/2$	-1/2	$-\sqrt{3}/2$
3	-1	0	1/2	$-\sqrt{3}/2$
4	-1/2	$-\sqrt{3}/2$	1	0
5	1/2	$-\sqrt{3}/2$	1/2	$\sqrt{3}/2$

of the hexagon in fig. 5.5  $ab, bc, \dots, fa$  respectively. Within a given segment, a sensor is  $\text{mod}(j, n)$  away from the hexagon vertices  $a, \dots, f$ . Define  $\Delta_j$  as  $j$  sensor's location relative to the vertices sensor in the segment that  $j$  belongs to.

The segment is the line connecting the hexagon vertices. All sensors will be on the segment, spaced out evenly on the segments. Since all the hexagons in the lattice are scaled versions of each other, we will proceed with the normalized hexagon with the vertices at distance of 1 to determine the location for sensors that are not at the vertices. The vertices are located at  $\mathbf{V} = [\cos(s\pi/3), \sin(s\pi/3)]$ ,  $s = 0, \dots, 5$ . Segments are defined by the vertices with the slope  $\mathbf{m} = [\cos((s+1)\pi/3) - \cos(s\pi/3), \sin((s+1)\pi/3) - \sin(s\pi/3)]$ . The location of the  $j$ -th sensor is on segment  $s$  weighted by  $\Delta$ . The location for the  $j$ -th sensor on the  $n$  ring of the lattice can be expressed as  $\mathbf{V} + \Delta\mathbf{m}$  with appropriate  $s$ . Table 5.2 lists  $\mathbf{V}$  and  $\mathbf{m}$  for each segment.

We will call the lattice with a sensor at the origin, described above, as  $\mathcal{L}_s$  with origin at  $o_s$ . The point farthest from any sensor,  $p_g$ , is located between 3 sensors (i.e. origin

in fig. 5.3.) We can characterize the distance between  $p_g$  and all sensors in the lattice by shifting the origin of the lattice by adding  $[0, d\sqrt{3}/3]$  to the location of all the sensors in the lattice. We will call the shifted lattice as  $\mathcal{L}_g$  with origin of the lattice at  $o_g$  in figure 5.5.

### **Bounding $g_n$**

We can bound the maximal utilities from geometric effect  $g_n$  for each ring of sensors by assuming all sensors in a given ring  $n$  are at the same distance to the shifted lattice origin  $o_g$  as the closest one in the ring. This will form the upper bound of the cooperation effect at each ring when we select the closest sensor for a given ring of sensors.

In the lattice where the void is at the origin, the closest point for each hexagon is at location  $g$  in fig. 5.5 because the lattice in  $\mathcal{L}_g$  is shifted *up*, hence the bottom edge  $ef$  is closer to the origin  $o_g$ . The vector of this closest edge  $o_gg$  is  $[0, -n\sqrt{3}/2] + [0, \sqrt{3}/3]$  for the  $n$ -th ring. The distance  $o_sg$  is

$$\begin{aligned} \|o_gg\| &= |\sqrt{3}/3 - n\sqrt{3}/2| \\ &= \left(n - \frac{2}{3}\right) \sqrt{3}/2 \end{aligned} \tag{5.6}$$

with  $n \geq 1$ .

### **Bounding cooperation utility**

Substitute  $g_n$  from eqn. 5.6 into eqn. 5.5, and from the fact that  $m_n = 6n$  for  $n \geq 1$ , we obtain the following

$$\begin{aligned}
k &= \sum_n \frac{6n}{[(n-2/3)\sqrt{3}/2]^\alpha} \\
&\leq c_0 \sum_n n^{-(\alpha-1)}
\end{aligned} \tag{5.7}$$

where  $c_0$  is some constant. By the integral test we can show  $k$  is bounded for  $\alpha > 2$  since  $\int_1^\infty n^{-(\alpha-1)} dn$  exists. For  $\alpha = 2$ , the rate of growth is logarithmic.

Therefore even allowing arbitrarily large scale cooperation, for  $\alpha > 2$ , the cooperation utilities will be finite. This bound, while not exact, shows the diminishing return in the total utility at the large cooperation scale. For the case of  $\alpha = 2$ , the above bound shows that the utilities increase only at a logarithmic rate with respect to the number of sensors cooperating. This result mirrors those of section 4.4.

## 5.5 Random deployment

In this section we present some simulation results of cooperation size and coverage area trade off using a random placement of sensors. Unlike the regular deployment in section 5.4, there is no guarantee that a random deployment can achieve complete coverage. Instead we will use coverage rate as the measure of utility. The scaling factor  $k$  is replaced with the number of sensors in the field as an indicator of how densely the sensors need to be placed.

The sensors are distributed uniformly in the field. A realization of one of the simulations is shown in figure 5.6. The threshold of coverage is the white area.

As seen in Fig. 5.7, the utility improvement saturated quickly as the number of sensors cooperating increases. The configurations of that simulation are as follows: randomly placing sensors according to a uniform distribution, declaring a point is covered when

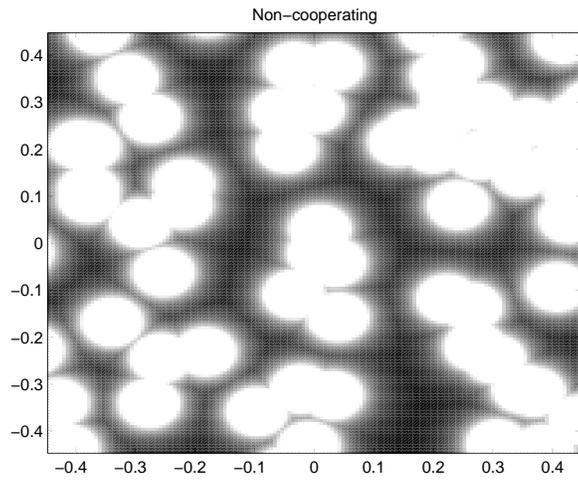


Figure 5.6: Example of deployment, no cooperation

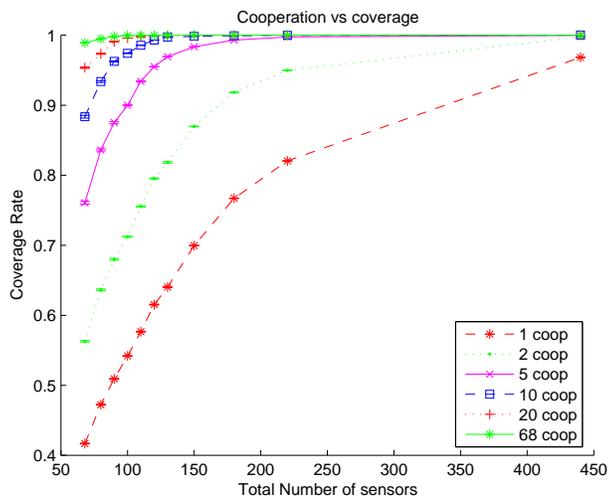


Figure 5.7: Utility - Cooperation - Sensors count

Table 5.3: Numerical example of cooperation size and deployment density trade off to achieve 95% coverage

Cooperation size	Sensors deployed
20	68
10	85
5	120
2	220

Table 5.4: Numerical example of cooperation size and deployment density trade off to achieve 90% coverage

Cooperation size	Sensors deployed
10	<68
5	100
2	150

the utility of the best group of cooperating sensors meets a threshold. Table 5.3 and 5.4 compare two numerical examples that illustrate that a larger cooperation scale or density is needed to achieve a higher threshold.

Recall from chapter 4 that at a fixed density large scale cooperation is bounded when  $\alpha > 2$  for a given point. The result is applicable to the random coverage problem here. Cooperation can provide only a finite benefit in terms of reduced density required. Conversely, whether cooperation is helpful depends on sensor density. At low density cooperation is helpful.

We can see that the geometry coupled with the fixed density is causing this behavior by the model used in sec. 4.4. The utilities contributed by far away sensors fall off

faster than the number of sensors increases as a function of distance. Therefore under the random deployment scenario we can also expect a diminishing return of cooperation similar to a lattice deployment strategy.

## 5.6 Conclusion

As we saw before in chapter 4, cooperation utilities provide significant utility improvement compared to no cooperation at all, but tend to provide only diminished returns with respect to large cooperation size. As we can see in fig. 5.7, 10 sensors cooperation yields similar performance to a cooperation size that is quite a bit larger. On the other hand, also seen in fig. 5.7, to achieve a given coverage rate, non-cooperation will require significantly more total resources.

Cooperation also enables us to reach a utility that would otherwise be unattainable. The cooperation size one should use is application dependent. As shown in fig. 5.4, at least in deterministic cases, the cooperation size and maximum attainable cooperation utility depend on  $\alpha$ .

## Chapter 6

# Reconstruction

### 6.1 Introduction

In this chapter, we will consider the problem of reconstruction from samples. The grand question is the trade off between the number of samples needed and the complexity necessary for the reconstruction technique. We will use several examples to demonstrate the trade off between the two. As expected, more samples allow for simpler reconstruction techniques and less samples are needed when more complex reconstruction is used. Fortunately, in some cases, the excess number of samples required for a simpler reconstruction technique is not necessarily much more than a more complex technique, for similar performance.

Simple reconstruction techniques tend to be more localized and require fewer assumptions, and thus are more robust. The trade off is that more samples will be needed. This can be thought of as a coverage problem, and simpler reconstruction techniques effectively reduce each sample's coverage of the source. More complex reconstruction, by making more assumptions about the source, effectively increases each sample's coverage. The trade

off is that those assumptions, in the form of a model, may be inaccurate to describe the source.

Many factors will drive the sampling strategy and reconstruction method. Factors such as the knowledge of the source, when the knowledge becomes available, difficulty of acquiring additional samples, and the dynamics of the source all contribute to this trade off. If detailed a priori knowledge of the source is available to both the sampling and the reconstruction entity, an accurate model based sampling and reconstruction technique can provide optimal performance with respect to sampling density and reconstruction accuracy. If the dynamics of the signal are slow, and acquiring additional samples is possible, an adaptive sampling technique can relax the prior knowledge requirement, and thus increase robustness. Even in the case of fast dynamics where an adaptive technique is not possible, knowing a model may enable a small number of observations to be sufficient to reconstruct the signal.

In this chapter, first we will use the well studied band limited signal as our source. From the Nyquist [13] and Shannon [14] sampling theorem, we know the theoretical minimal number of samples necessary to achieve perfect reconstruction. We will compare that with linear interpolation and spline reconstruction [46], [47]. We will run an experiment to observe their performance.

Next we will introduce a few step discontinuities to the source. This modification will demonstrate the sensitivity of model specific reconstruction when a mismatch is introduced. Here we will show how a modification to the spline can help us recover from the discontinuity. An adaptive sampling technique is used to find the location of the dis-

continuity and a decision based on a few adjacent samples will be needed to estimate the location of the discontinuity. We can estimate if a sample crosses the discontinuity with the Expectation-maximization (EM) algorithm [31]. This demonstrates that local cooperation remains viable if the modification to the baseline model is known.

We will then replace the band limited source with a source that consists of a few discrete tones with unknown frequency and phase. The frequencies of the tones will be drawn randomly, and allowed to be very high. Compressed sensing [48] can reconstruct this type of signal with a significantly fewer number of samples compared to conventional techniques. Using the sparse in frequency domain model, compressed sensing can accomplish this reconstruction using more complex algorithms with less samples. We will then add discontinuities to the source, and compare reconstruction quality with number of samples needed. The wavelet transform, which can represent discontinuities in a smooth signal efficiently, will serve as a bench mark in that section.

In this chapter we will see that local cooperation can be competitive with respect to global cooperation in terms of resource use in some situations, but this is not universal. The local cooperation strategy also has the inherent advantage of limiting local model mismatch to the locality where the mismatch occurs. For some other situations such as sparse sources, global cooperation such as compressed sensing reconstruction provides superior utility with a significantly reduced number of samples.

## 6.2 Source and Classical reconstruction technique

We will consider the band limited signal as the source for the reconstruction problem. We obtain the band limited signal by band pass filtering of white Gaussian noise. We will consider the following methods of reconstruction and compare their performance in reconstruction quality and number of samples needed. The quality of reconstruction is measured by the mean of the squared difference between the reconstructed signal  $\hat{x}(t)$  and the source  $x(t)$ .

The reconstruction problem can be stated as follows. Source  $x(t)$  is sampled with  $y_i = x(t_i)$ . The reconstruction algorithm is  $f(Y)$  where  $Y$  contains several samples and produces  $\hat{x}(t) = f(Y)$ .

### 6.2.1 Nyquist sampling

A signal  $x(t)$  is said to be band limited if the Fourier transform of the signal

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft} dt \quad (6.1)$$

is 0 for all  $|f| > B$  and  $B$  is the highest frequency of the signal  $x(t)$ . From the Nyquist/Shannon sampling theorem, samples  $1/2B$  apart will be sufficient to reconstruct the signal perfectly using a sinc function to interpolate between samples [14]. A sinc function  $s(t)$  is defined as

$$s(t) = \frac{\sin(\pi t)}{\pi t} \quad (6.2)$$

The Nyquist sampling theorem establishes the minimum number of samples necessary to completely recover the signal. The reconstruction depends on every sample because the sinc function exists between  $-\infty$  to  $\infty$ .

### 6.2.2 Nearest sample

The simplest reconstruction technique is to assume the value at any other location is identical to the nearest sample. If sampled at sufficient density, this simple technique will meet the quality criteria. However, the density may be very high compared to other reconstruction techniques.

### 6.2.3 Linear Interpolation

Instead of reconstructing using a sinc function, a linear interpolation between two samples may also be used. This approach limits a sample's contribution to the region between only two samples. For example, the interpolation between  $\hat{x}(t_i)$  and  $\hat{x}(t_j)$  from samples  $t_i$  to  $t_j$  is

$$\begin{aligned}\hat{x}(t) &= m_i x + b_i, t_i \leq t \leq t_j \\ \text{where } m_i &= \frac{y_j - y_i}{x_j - x_i}, \\ b_i &= y_i - m_i t_i\end{aligned}\tag{6.3}$$

By taking this similar reconstruction approach, more samples will be necessary to achieve the desired quality unless the source is also piecewise linear and sampled at the end points of each line segment.

### 6.2.4 Spline Interpolation

We can improve upon linear interpolation by using a higher order polynomial that takes more samples locally to reconstruct. For comparison we will use the cubic spline, which is a third-order polynomial, for reconstruction. [46] contains additional theoretical

background on splines.

Basically, the idea is to go one step beyond the piecewise linear approximation to obtain a smoother reconstruction. Using a piecewise polynomial of higher order is necessary if an additional refinement such as smoothness is desired. A popular choice is to use a cubic approximation as the model of interpolation. Spline interpolation assumes that both the first and second derivative are continuous.

### **6.2.5 Performance comparison**

Simulations to compare the performance of the above reconstruction techniques were performed. The simulation consists of 100 trial runs of generating the band limited signal, sampling at regular intervals, and then performing reconstruction based on the methods above. An instance of the reconstruction and sampling is shown in figure 6.1. A reconstruction is declared a success when the mean square error is less than  $2 \times 10^{-5}$ .

Summary results presented in figure 6.2 show the performance change as the number of samples increases. As we can see in the figure, the spline performs relatively well. Linear reconstruction takes about 4 times the number of samples. Reconstruction using the nearest sampled value for a band limited source is unsuccessful. This simulation reinforces the conclusion that local cooperation at a small scale can provide most of the performance. No cooperation, as in the case of nearest value only will incur a great performance penalty.

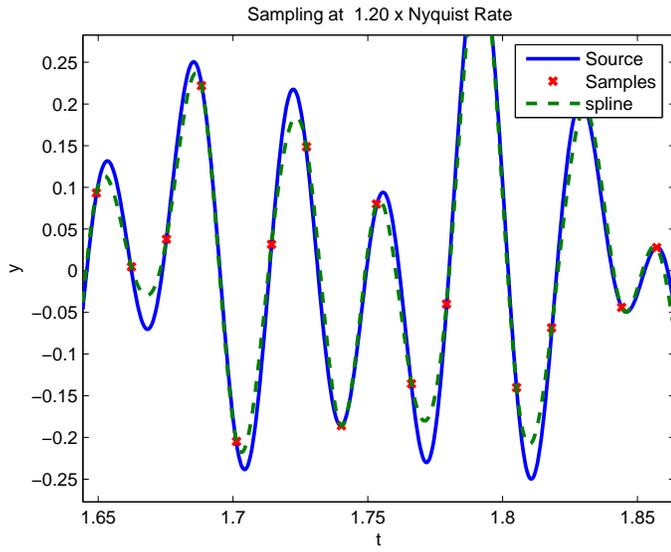


Figure 6.1: Reconstruction example

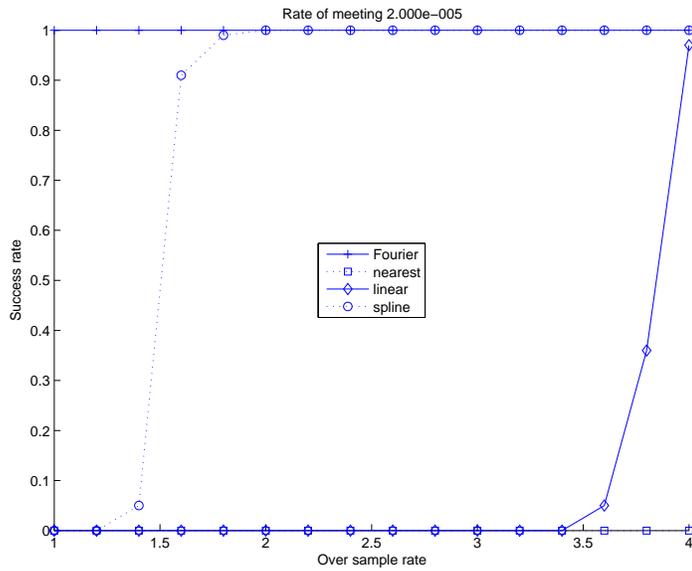


Figure 6.2: Performance comparison between reconstruction method and resources trade-off

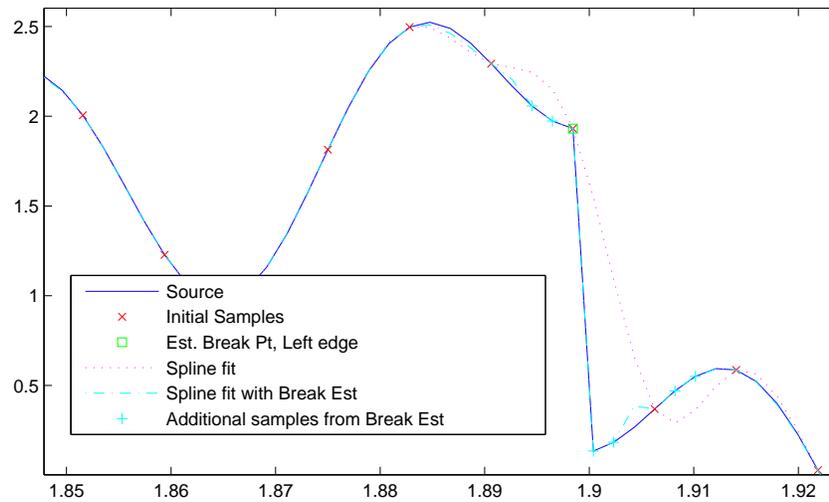


Figure 6.3: Break point example

### 6.3 Modifying the source

In this section we will investigate the impact of model mismatch to cooperation size by modifying the source. Specifically, we will introduce several steps in the signal as follows:

1. Randomly select  $d$  points in the source according to the uniform distribution as the location for a discontinuity
2. Randomly select the step direction, either  $+s$  or  $-s$  with a probability of  $p$  for the direction
3. Shift the source in between a pair of selected points by the direction selected in previous step.

Figure 6.3 shows one instance of this step.

By introducing this type of discontinuity, we added many high frequency components to the source. Fig. 6.3 also shows the result obtained with spline reconstruction used in the previous section.

The discontinuity introduced is a local phenomenon. If we can estimate the location of the discontinuity, then we can apply the results in the previous section by excluding the discontinuity itself.

### 6.3.1 Determine where the breaks are

Adaptive sampling techniques will allow us to obtain additional samples near the discontinuity in a discrete manner without significantly raising the overall sampling rate. Multiple iterations of sampling will be performed. At each iteration, sample points are decided based on the previous iteration. The problem then is how to decide if points between two sampled point contain the step introduced by the discontinuity. Once a boundary is estimated, using the bisection approach we can quickly ( $\mathcal{O}(\lceil(u-l)/\epsilon\rceil)$  [33]) narrow down the location of the steps.  $u, l$  are the upper and lower boundary respectively where a break at  $p^*$  is present, and  $\epsilon$  is the allowable uncertainty of the boundary at the end of the bisection procedure.

The bisection we will use can be described as follows [33]:

**given**  $l \leq p^*, u \geq p^*$  tolerance  $\epsilon > 0$

**repeat**

1.  $t := (l + u)/2$
2. Compute the slope  $m_l = (x(t) - x(l))/(t - l)$  and  $m_u = (x(t) - x(u))/(t - u)$

3. **if**  $|m_l| > |m_u|$ ,  $u := t$ ;      **else**  $l := t$ .

**until**  $u - l < \epsilon$

### 6.3.2 Finding the breaks in the first place

Using the above procedure we can quickly find the break point if an initial boundary is given. The next question is how to decide if a break occurs between two sampled points. The discontinuity introduced leads to a step size of  $s$ . A sufficiently large step  $s$  will introduce a large derivative compared to the source derivative. Therefore by comparing the derivative between the samples and its neighbor we can estimate if a given sample is the closest sample to a discontinuity. This collection of derivatives and comparisons can also be performed at a local scale. The cooperation size locally will determine the prevailing ‘typical’ derivative.

To perform this estimation of the boundary, we will use the expectation maximization (EM) algorithm [31] to separate the derivative between two classes using only local data.

The EM algorithm is an iterative algorithm consisting of two parts:

1. Compute the conditional expectation of the log-likelihood, and
2. Set the likelihood of each distribution equal to the value that maximizes the log-likelihood.

We will run the absolute value of the difference between samples

$$|x(t_{i+1}) - x(t_i)| \tag{6.4}$$

through the EM algorithm to identify the samples that are adjacent to the break. Using absolute value simplifies the initialization because we assign the distribution with the larger mean a smaller prior probability, as this distribution represents the discontinuity.

The EM algorithm will provide a distribution that consists of a mixture of different distributions and their weightings. In turn we will use that to decide if a given point is likely to be an edge of the discontinuity. This point will most likely NOT be the actual break point, but merely the closest point in the present sample set. With the selection of the derivative calculation used in 6.4, the matching index will be the left boundary of the discontinuity. We will start the bisection algorithm described in section 6.3.1 with sample point  $x(t_l)$  as the left edge and  $x(t_l + 1)$  as the right edge if  $|x'(t_l)|$  is declared to be the left boundary of the discontinuity by the EM algorithm.

### 6.3.3 Simulation result

Using the simulation frame work in section 6.2, we added the discontinuity and a mechanism that handles the discontinuity. We continue to use the spline reconstruction algorithm and use mse of  $2^{-5}$  as the pass/fail criterion of the reconstruction for a given trial. The result of this reconstruction exercise is shown in figure 6.4. To see how the discontinuity impacts our performance, we will also monitor the performance of the QoS but dropping the worst 0.1% of the sources, and declaring those last 0.1% of points as outage. Figure 6.4 shows that the few worst case points dominate the performance of the overall QoS.

In figure 6.5 we turn around and ask how many of the sources have to be declared as an outage to meet the mse criteria. The connected line shows the mean of the outage rate and the error bar denotes the maximum and the minimum outage rate in the

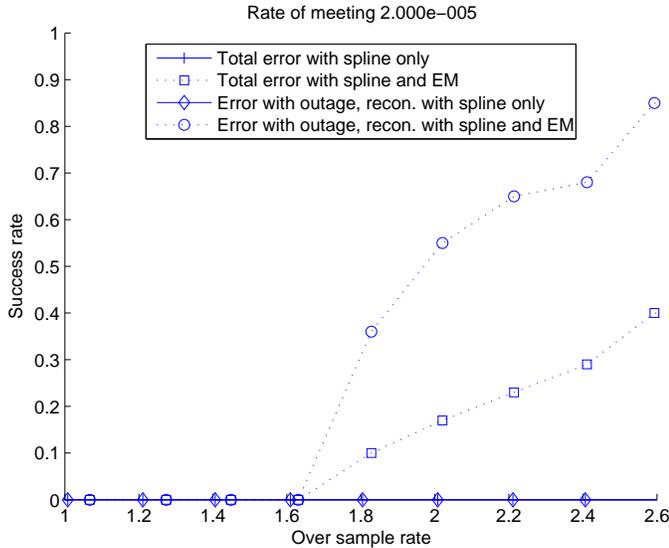


Figure 6.4: Rate of meeting the QoS for a given initial sampling rate

experiment. Here once again the few points continue to dominate the overall reconstruction mse. As expected, spline with EM performs better than reconstruction with spline only for a given initial sampling rate. That is because EM mitigated errors near the discontinuities. Discontinuities require high density sampling. Without the outage relaxation we must sample everywhere densely. However, as figure 6.5 shows, it is the very sharp discontinuities that dominate overall performance. Note the mean outage rate under spline with EM reconstruction is near 0, with the worst cases outage around 1.5%.

## 6.4 Trade off of sample size and reconstruction complexity

We will explore the trade off between sample size and reconstruction complexity with a case study. The modified source we used in section 6.3 can be represented with few coefficients by a wavelet transform. Wavelet transforms can describe the discontinuities

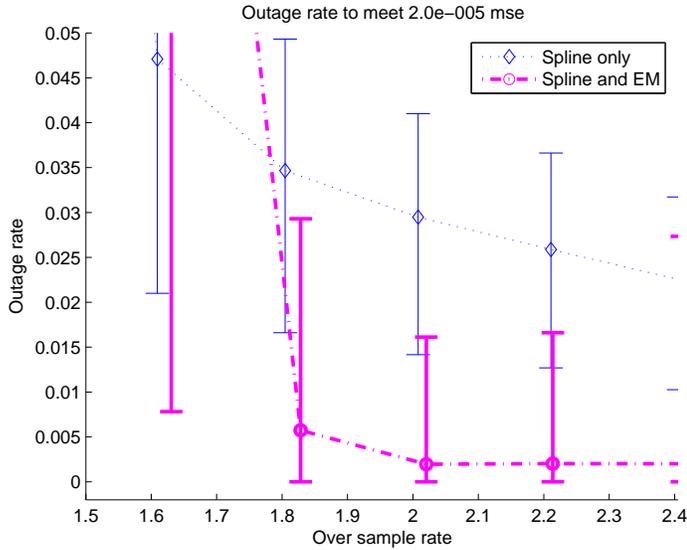


Figure 6.5: Outage rate to meeting QoS for a given initial sampling rate

accurately while retaining a compact description of the smooth part of the source. The wavelet transform takes a dense sampling of the source to produce the final compact representation of the source. Compressed sensing on the other hand uses a few samples of the waveform to reconstruct a source accurately under the right condition. If the source and model mismatch in certain aspects, compressed sensing breaks down. Spline reconstruction with EM used in section 6.3 falls somewhere in between the two methods, in that relatively few samples are taken (as in compressed sensing) but with a somewhat less compact representation than using wavelets.

#### 6.4.1 Wavelet Compression

Wavelet analysis breaks down the signal into multiple level of resolution. With this multiple resolution wavelet analysis can detect and handle discontinuities much more

efficiently than the Fourier transform. Additionally, by using different wavelets as basis functions, different local properties of the source can be emphasized.

A wavelet transform, instead of using a complex exponential function as the basis, uses a wavelet basis  $\Psi(s, p, t)$  to perform the transformation. Scale  $s$  describes the stretching or compression of the wavelet basis and conceptually is similar to frequency in the Fourier transform. Position  $p$  is similar to phase in the Fourier transform. It describes where features of the source are located. A continuous wavelet transform of  $f(t)$  is computed by

$$C(s, p) = \int_{-\infty}^{\infty} f(t)\Psi(s, p, t)dt \quad (6.5)$$

By stretching and compressing the wavelet, instead of only varying frequency, the wavelet transform essentially introduces different time scales in the transformed domain. In fig. 6.6 is a representation of the Fourier transform domain. Each frequency bin is present in time between  $-\infty$  and  $\infty$ . In fig. 6.7 a similar representation for the wavelet transform is presented. The wavelet transform does not require a given scale of wavelet to be present at all positions. A windowing of the source is present. This allows the wavelet to handle discontinuities more efficiently compared to the Fourier transform.

Similarly to the discrete Fourier transform, there is a discrete implementation of the wavelet transform. An efficient way to perform the wavelet transform was presented in [49] where the wavelet transform is implemented as a sequence of high pass and low pass filtering, followed by a down-sampling operation.

We can describe the discontinuities that we introduced with a compression wavelet located at the position of the discontinuity in the source. The Haar wavelet, which is

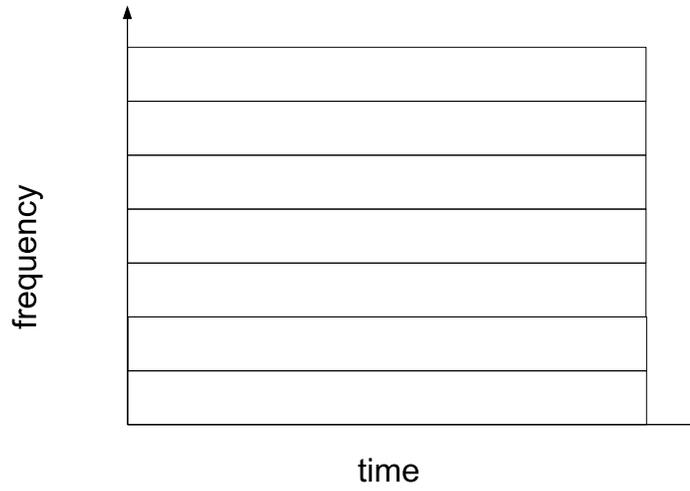


Figure 6.6: Fourier Transform domain

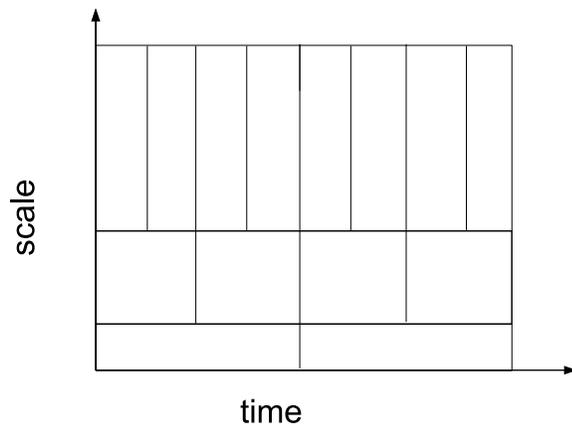


Figure 6.7: Wavelet Transform domain

basically a step, can describe the discontinuity we introduced to the signal in section 6.3 with a few parameters. On the other hand, the Fourier transform requires a large number of parameters to describe the same type of discontinuity.

### 6.4.2 Compressed Sensing

The compressed sensing problem can be expressed as follows: suppose a source  $f \in \mathbb{R}^n$  is  $s$  sparse in some orthonormal basis  $\Psi = [\psi_1 \psi_2 \dots \psi_n]$

$$f(t) = \sum_{i=1}^n x_i \psi_i(t)$$

where  $x_i$  is the coefficient of  $f(t)$  in basis  $\psi_i$ . Then with  $m$  observations  $y$  of  $f(t)$ , each observation made with  $\phi_k(t)$ ,  $k = 1, \dots, m$ , we have

$$y_k = \langle f, \phi_k \rangle \tag{6.6}$$

All of the observations  $y_k$  are not merely time sampled values of  $f$  as in the classical sampling and reconstruction. Instead, each observation, as stated in the eqn. 6.6, consists of the inner product between the source and a column of  $\Phi$ . In other words, each observation used by the reconstruction is a combination of time sampled values of  $f$ . However, a column of  $\Phi$  can be all 0, i.e. a time instance that is not sampled at all. The reconstruction process does *not* require all time instances to be sampled. This allows for reconstruction with a significantly fewer number of time samples. We do not need to form this  $\Phi$  entirely at the sampler either. We only need the time sample of  $f$  where a corresponding column in  $\Phi$  contains a non-zero entry. In our example below many of those columns are in fact all zero. When  $\Phi = [\phi_1; \phi_2; \dots \phi_m]$  and  $\Psi$  satisfies the compressed sensing condition we can recover

$f$  with  $m = \mathcal{O}(s \log(N/s))$  observations[50]. The observations use by compressed sensing can be thought of as a parallel bank of random sequence ‘matched filters’[48].

[48] along with the rest of the issue of IEEE Signal Processing Magazine in March 2008 provided a good overview of compressed sensing. [51, 52, 53, 54] are a few references that establish the theoretical basis of compressed sensing. Extension to a continuous sparsity model is discussed in [55]. Using compressed sensing to detect sparse signals in wide bandwidth was presented in [56, 57].

Techniques to sample below the Nyquist rate and reconstruct had been attempted previously. By studying the signal structure, one can reconstruct perfectly with a sampling rate below the Nyquist rate. The difference between the prior sub-Nyquist sampling techniques and the compressed sensing are that in the prior techniques the sampler needs to know or be adaptive to the signal parameters, while in compressed sensing randomness is used to handle the variation in signal parameters. In [58], direct sequence CDMA signals were sampled below the chip rate. Speech signals were sampled below Nyquist rate in [59]. Channel state estimates using subspace techniques allowing sampling below the Nyquist rate were presented in [60]. Using the subspace technique also allowed signal reconstruction with sampling rate below the Nyquist rate when the signal rate of innovation is finite [61]. A signal rate of innovation corresponds to degrees of freedom per unit time [62]. Another strategy is using non uniform sampling [63, 64, 65]. Non uniform sampling is applicable when the signal is multi band, i.e. energy is not flat across the entire spectrum. For non-uniform sampling to work, we also will need to learn of the concentration of the signal either a priori or adaptively at the sampler.

[52] established that with a random basis such as i.i.d. Gaussian or Bernoulli with probability of 0.5 of  $\pm 1$  as elements of  $\Phi$ , solving the optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^N} \quad & \|x\|_1 \\ \text{s.t.} \quad & \Phi x = y \end{aligned} \tag{6.7}$$

will recover  $x$  exactly. Therefore we can reconstruct the source in the sparse domain from observations by solving the convex optimization problem in eqn. 6.7, which can be solved efficiently [33]. Alternatively, the reconstruction problem can be solved by an iterative algorithm CoSaMP: Compressive Sampling Matching Pursuit [66]. CoSaMP is a ‘belief propagation’ algorithm that iteratively performs the following steps:

1. Forming a proxy to the signals residual,
2. Selecting the largest components within the proxy,
3. Merge the selection with the previous selection,
4. Solve a least squares problem with the merged proxy
5. Prune the solution of the least squares problem down to the sparsity size
6. Update the signal residual

Section 2 of [66] describes the algorithm in detail. The complexity of each iteration is  $\mathcal{O}(mN)$  [66].

Clearly, the reconstructions are more complex when compared to classical techniques. Reconstruction of signals with classical techniques such as interpolation is simpler compared to solving eqn. 6.7. In the classical approach, the communication between sampling and reconstruction is typically handled separately. In particular, the samples are

compressed down from the sampling rate to the necessary reconstruction rate and then transmitted. That implies some samples may be discarded, if the sampling rate is higher than the reconstruction rate. The reconstruction rate depends heavily on the reconstruction method and the source structure.

One of the advantages of compressed sensing is that the sampling operation does *not* need to know the transform  $\Psi$  necessary to map  $f$  to the sparse representation. Only the reconstruction operation needs to know that transformation if  $f$  itself is not sparse. All the sampling operation needs to provide are the  $m$  measurements and the  $\Phi$  used to generate the measurements. Additionally, the value of  $m$  needed for reconstruction depends on the incoherences between  $\Phi$  and  $\Psi$ . Coherence between  $\Phi$  and  $\Psi$  is defined as [48]

$$\mu(\Phi, \Psi) = \sqrt{(n)} \max_{1 \leq k, j \leq n} |\langle \phi_k, \psi_j \rangle| \quad (6.8)$$

Smaller  $\mu$  implies less coherence between  $\Phi$  and  $\Psi$ , which in turn implies fewer measurements are needed.

The implication of incoherency between the measurement matrix  $\Phi$  and the sparse representation  $\Psi$  is that each measurement obtained will have some contribution from each of the sparse components. This property enables the solution to the optimization problem of eqn. 6.7 to recover the sparse representation  $x$  [52].

Although most signals in real life are not strictly sparse, compressed sensing also applies to compressible signals where the sorted magnitude of coefficients  $x_i$  decay quickly. [52] uses power law decay, and shows compressed sensing results hold in those cases.

Compressed sensing can also be viewed as a compression technique, compressing the source to the sparse representation via the measurement matrix  $\Phi$ . Thanks to the

randomness nature of  $\Phi$ , compressed sensing is nearly universal[52], since it is very unlikely for a structured  $\Psi$  to be coherent with a randomly generated  $\Phi$ . However, in order to deploy compressed sensing effectively, one must verify the sparsity of the signal. While one can always increase the dimensions of the observation until the signal becomes sparse relative to the dimensions of observation, classical techniques in lower dimensions in some cases may provide a more effective solution, as we will see in section 6.4.4.

The parameter in compressed sensing is the sparsity of the source. The sparsity of the source determines the number of observations the reconstruction needs. Or in other words the number of adaptations available at the sampler is limited to the number of measurements the reconstruction can use. For a strictly sparse signal, this is not very interesting other than determining if we can reconstruct perfectly or not at all. For a compressible signal, this adaptation in effect controls the reconstructed signals SNR[67].

### **An Example of Compressed Sensing**

We will use a source that is sparse in the frequency domain and measure samples in the time domain. We intentionally placed a high frequency component in the signal. With conventional sampling, the high frequency component will require dense sampling. With compressed sensing, the example shown in Fig. 6.8 only requires  $m = 60$  samples to reconstruct the  $n = 1024$  points waveform.

We will also add discontinuities to the source, similar to the previous section. When discontinuities are introduced to the source, the signal is no longer strictly sparse, but remain compressible and therefore compressed sensing still applies. In Fig. 6.9,  $m = 260$  samples are needed, to achieve a mean square error of 0.0101.

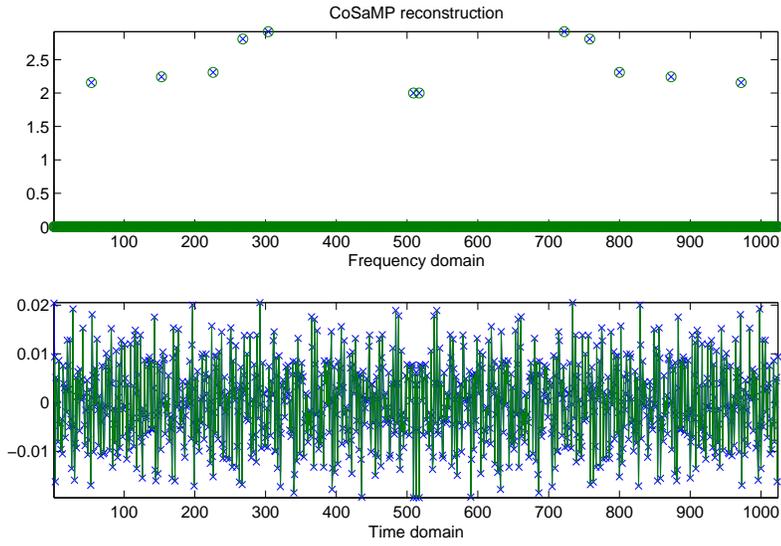


Figure 6.8: Sparse in frequency domain

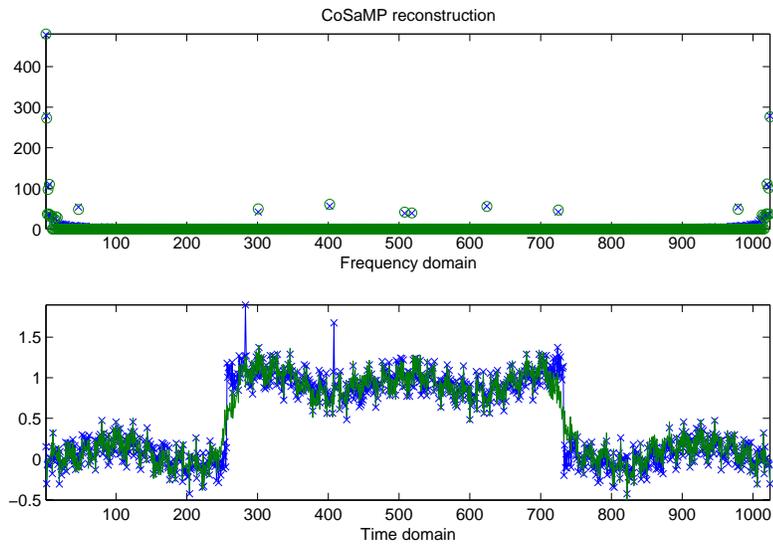


Figure 6.9: Sparse frequency with discrete discontinuities

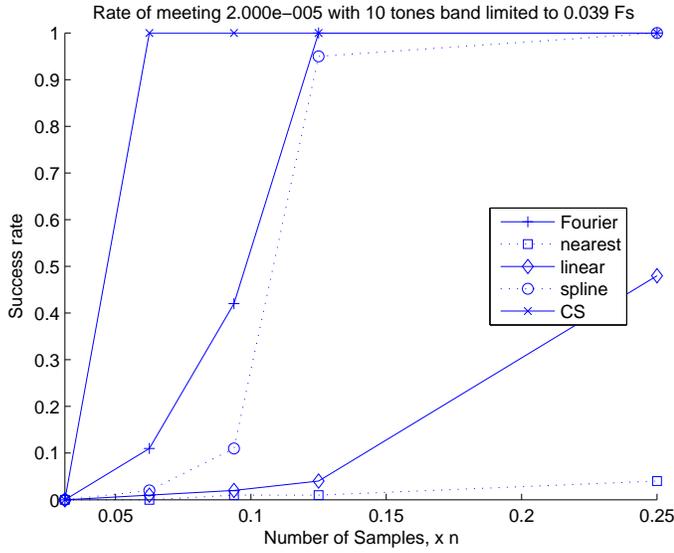


Figure 6.10: Comparing compressed sensing with classical reconstruction

### 6.4.3 Comparing Compressed Sensing with classical technique

We will next illustrate the different strengths of the compressed sensing technique and the classical reconstruction technique. We will use a source similar to those in section 6.2, but replace the band limited noise source with multiple discrete tones. The frequency for each tone is drawn randomly according to a uniform distribution between  $0.5$  to  $0.039F_s$ . The number of samples required for perfect reconstruction under the Nyquist-Shannon sampling theorem for this source is up to  $0.078 \times n$ . Figure 6.10 shows the success rate of meeting the mse of  $2 \times 10^{-5}$  by various reconstruction techniques, similar to fig. 6.1. Compressed sensing performed better when the number of observations is below the Nyquist-Shannon Sampling theorem requirement. As the number of observations grows beyond the minimum required by the sampling theorem, the classical techniques are able to meet the mse goal as in section 6.2.

The advantage of compressed sensing is clear when the signal is sparse with potential high frequency components. Compared to the previous solution of sub-Nyquist sampling, we have more flexibility with compressed sensing because we can operate with different sparsity transformations *without* any change in sampling strategy. However, if the signal is non sparse the classical techniques will perform better.

#### **6.4.4 Comparing Compressed Sensing with classical technique with modified model, using wavelet representation as reference**

Similarly, a comparison of performance between compressed sensing and reconstruction techniques in section 6.3 was performed when the signal contains steps. With this type of source, the spline reconstruction with EM is more robust compared to compressed sensing. In particular, the discontinuities caused large amounts of artifacts in the frequency domain which in turn breaks the sparsity model assumption used in this simulation. Fig. 6.12 shows one example of compressed sensing reconstruction of this type of signal. There are a large number of residuals in the frequency domain disregarded by the compressed sensing reconstruction. Although the reconstruction is able to capture the majority of the trend of the signal, splines with the EM algorithm outperform compressed sensing in this regime where the maximum frequency is not too high. The discontinuities caused difficulty for the Fourier transform from the frequency domain to the time domain because the frequency domain is not strictly sparse under the Fourier transform.

The spline reconstruction with EM performs well given the number of samples it used. From the wavelet compression reference, our spline with EM performs close to the wavelet compression performance given a number of observation similar to the number of

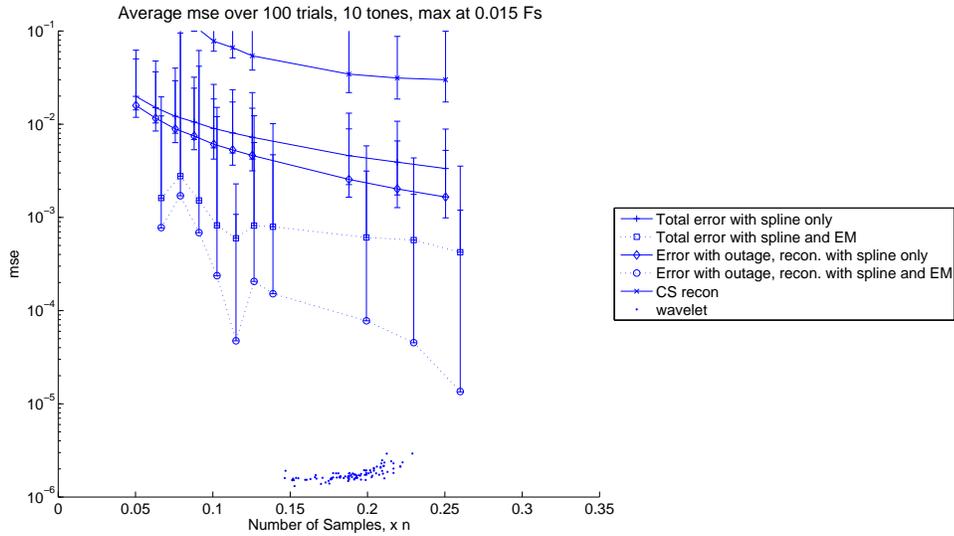


Figure 6.11: Comparing compressed sensing with classical reconstruction and steps in signal non-zero coefficient in the wavelet representation. In other words, the spline with EM made good use of the samples provided with only a local approach.

As in the smoothed signal case in sec. 6.4.3, compressed sensing outperforms the classical technique only when the source is sparse but with tones at high frequency. However, for the compressed sensing to have significant advantage, the highest frequency needs to be higher in the case with discontinuities. In the following we increase the highest frequency limit to  $0.146F_s$  from  $0.039F_s$ . Instead of showing the success rate of meeting a mse of  $2 \times 10^{-5}$ , the mse statistics are shown in fig. 6.13. When the number of samples is small, the conventional technique outperforms compressed sensing. However, compressed sensing can identify the fundamental tones present in the signal with less observations compared to classical techniques. Thus compressed sensing performs better when the maximum possible tone frequency is high. The error bars show the maximum and the minimum reconstruction error mse in the 500 trials.

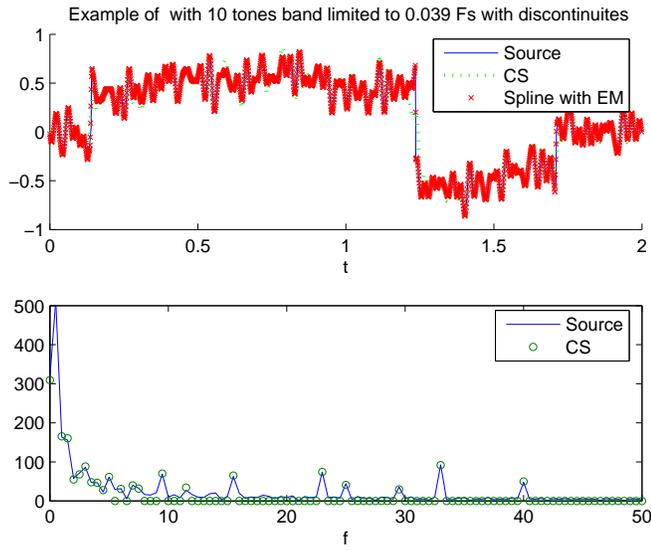


Figure 6.12: Reconstruction examples, Spline with EM and CS

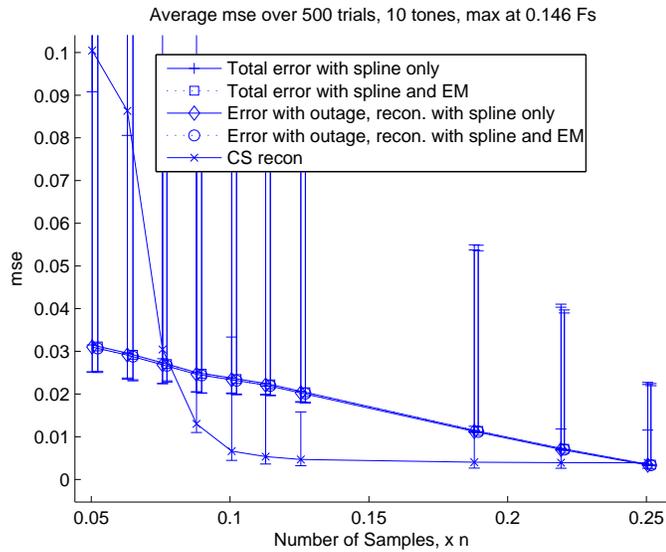


Figure 6.13: Comparing compressed sensing with classical reconstruction and steps in signal, high signal frequency

In [50], wavelet transforms were used as the sparse transform in the compressed sensing. By using a more accurate model, compressed sensing can be more competitive compared to what we have seen above. This illustrates the importance of an accurate model. Inaccurate models can degrade reconstruction quality as for the case of using the Fourier transform as the basis for compressed sensing on signals with discontinuities. With an accurate model local methods such as the spline can be competitive with respect to global methods such as wavelet compression.

## 6.5 Conclusion

In the classical reconstruction problem, a few samples cooperating can provide a simplified reconstruction method with high quality at a low cost. In addition, a change in the source may only require a minor modification to the reconstruction technique to handle the changes of the source. In these cases massive global cooperation is not necessary. Instead, an adaptive sampling technique guided by local statistical inference such as the EM algorithm is sufficient to handle the modified source. However, when the source model is changed significantly, in our example from band limited to a sparse source, global cooperation can significantly reduce the number of samples necessary for reconstruction to meet a desired QoS. Nevertheless global cooperation with an inaccurate model cannot compete against local cooperation with more a accurate model.

## Chapter 7

# Conclusion

From the previous chapters we saw that small, local scale cooperation provides noticeably improved quality of service such as improved coverage area, mitigation of fading outage, or reduced reconstruction error in the sensor. On the other hand there is little benefit in having a large number of sensors cooperating. This in turn simplifies cooperation algorithms because we only need to consider a small scale of cooperation.

From the localization example in chapter 3, we saw that it is beneficial to use a more complex sensor selection algorithm to determine the sensor set. The small scale nature of the problem limits the complexity, making the more complex algorithm problem tractable, and at the same time the more complex algorithm quickly extracts most of the utility obtainable within the sensor network. With this result, we know that most of the overall utilities are derived from the first few sensors. Therefore we should focus on selecting the first few sensors using techniques discussed in [1, 2, 3, 4].

In chapter 4 we saw that a distance loss model with fixed sensor density places a

limit on the total utility possible with cooperation. Local cooperation provides a noticeable benefit, but global cooperation provides only limited improvement. Thus a small scale of cooperation to mitigate channel degradation should be part of the design consideration of a sensor network[5, 6], but a large scale, global cooperation is not necessary.

Similar effects can be seen in chapter 5 with respect to the coverage problem. We have shown that cooperation can help bridge small gaps in coverage, but cooperation cannot be used to cover arbitrarily large gaps. The coverage is an improvement over non-cooperating sensors such as modeled in [7, 8, 9] because the gap can be filled via local cooperation.

In the reconstruction problem, similar results are observed, as we saw in chapter 6. We showed that while local reconstruction algorithms use slightly more resources compared to the optimal global algorithm, the local algorithm is less sensitive to model error. While wavelets can provide a very compact description of the source with high quality reconstructed solutions, we must sample densely at first. On the other hand, compressed sensing may not give a satisfactory reconstructed solution if the model used in the reconstruction deviates too much from the actual model of the source. Local algorithms such as spline reconstruction offer something in between in terms of sample size and reconstruction quality. In addition, local algorithms are simple to modify to include adaptive sampling.

From the examples presented in this thesis, we can see that cooperation at a local scale provides great improvement over non-cooperative modes of operation. Large scale global cooperation in most cases provides only marginal benefit. This limitation depends on the global relationship between individual sensors or samples. Only when the global

model is accurate could global cooperation be attempted.

An interesting question is how to verify model accuracy. All cooperation depends on the model that relates different observations together. Without a sufficiently accurate model cooperation is impossible. The accuracy required is an application specific requirement. A more flexible model enables more adaptable global algorithms, such as combining the wavelet transform with compressed sensing. However, there is a trade off between quality and complexity in implementation.

# Bibliography

- [1] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, “Robust submodular observation selection,” *Journal of Machine Learning Research*, vol. 9, pp. 2761–2801, 2008.
- [2] H. Wang, K. Yao, G. Pottie, and D. Estrin, “Entropy-based sensor selection heuristic for target localization,” in *Information Processing in Sensor Networks (IPSN’04)*, Berkeley, CA, Apr. 2004.
- [3] J. Byers and G. Nasser, “Utility-based decision-making in wireless sensor networks,” in *MobiHoc ’00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. Piscataway, NJ, USA: IEEE Press, 2000, pp. 143–144.
- [4] P. V. Pahalawatta, T. N. Pappas, and A. K. Katsaggelos, “Optimal sensor selection for video-based target tracking in a wireless sensor network,” in *International conference on Image Processing, ICIP*, Singapore, Oct. 2004.
- [5] M. D. Yacoub, *Foundations of Mobile Radio Engineering*. CRC Press, 1993.
- [6] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.
- [7] S. Adlakha and M. Srivastava, “Critical density thresholds for coverage in wireless

- sensor networks,” in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, March 2003, pp. 1615–1620 vol.3.
- [8] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Worst and best-case coverage in sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 4, no. 1, pp. 84–92, Jan.-Feb. 2005.
- [9] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2001, pp. 1380–1387 vol.3.
- [10] W. Wang, V. Srinivasan, B. Wang, and K.-C. Chua, “Coverage for target localization in wireless sensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 7, no. 2, pp. 667–676, February 2008.
- [11] D. Wang, Q. Zhang, and J. Liu, “The self-protection problem in wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 3, no. 4, p. 20, 2007.
- [12] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar, “Redundancy and coverage detection in sensor networks,” *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 94–128, 2006.
- [13] H. Nyquist, “Certain factors affecting telegraph speed,” *Bell Systems Technical Journal*, vol. 3, p. 324, 1924.
- [14] C. Shannon, “Communication in the presence of noise,” *Proc. IRE*, vol. 37, pp. 10–21, 1949.

- [15] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug 2001.
- [16] H. Durrant-Whyte, "Where am i? a tutorial on mobile vehicle localization," *Industrial Robot: An International Journal*, vol. 21, no. 2, pp. 11–16, 1994.
- [17] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.
- [18] A. Murillo, J. Guerrero, and C. Sagues, "Robot and landmark localization using scene planes and the 1d trifocal tensor," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct. 2006, pp. 2070–2075.
- [19] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, Oct. 2003, pp. 1403–1410 vol.2.
- [20] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *IEEE INFOCOM*, Miami, FL, March 2005.
- [21] H. L. V. Trees, *Detection, Estimation, and Modulation Theory Part I*. John Wiley & Sons, 2001.
- [22] F. Bian, D. Kempe, and R. Govindan, "Utility-based sensor selection," in *Information Processing in Sensor Networks (IPSN'06)*, Nashville, TN, Apr. 2006.

- [23] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, ser. Information and System Sciences Series. Prentice Hall, 2000.
- [24] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [25] N. Balakrishnan and A. C. Cohen, *Order Statistics and Inference*. Academic Press, 1991.
- [26] E. W. Weisstein, “Circle packing,” From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/CirclePacking.html>.
- [27] P. Gupta and P. Kumar, “The capacity of wireless networks,” *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, Mar 2000.
- [28] A. Graps, “An introduction to wavelets,” *Computational Science & Engineering, IEEE*, vol. 2, no. 2, pp. 50–61, Summer 1995.
- [29] D. Lee, “Jpeg 2000: Retrospective and new developments,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 32–41, Jan. 2005.
- [30] E. Dyer, “Compressive sensing resources — rice dsp,” <http://www.dsp.ece.rice.edu/cs>.
- [31] J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning*. Springer-Verlag, 2001.
- [32] V. Isler and R. Bajcsy, “The sensor selection problem for bounded uncertainty sensing models,” in *Information Processing in Sensor Networks (IPSN’05)*, Los Angeles, CA, Apr. 2005.

- [33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, 2004.
- [34] A. Savvides, “Design implementation and analysis of ad-hoc localization methods,” Ph.D. dissertation, University of California, Los Angeles, 2003.
- [35] N. Patwari, “Location estimation in sensor networks,” Ph.D. dissertation, University of Michigan, 2005.
- [36] C. Chang and A. Sahai, “Estimation bounds for localization,” in *IEEE Conference on Sensor and Ad Hoc Communications and Networks*, Oct. 2004.
- [37] J. C. Chen, K. Yao, T. L. Tung, C. W. Reed, and D. Chen, “Source localization and tracking of a wideband source using a randomly distributed beamforming sensor array,” *The International Journal of High Performance Computing Applications*, vol. 16, pp. 259–272, Fall 2002.
- [38] N. Ahmed, S. S. Kanhere, and S. Jha, “Probabilistic coverage in wireless sensor networks,” *Local Computer Networks, Annual IEEE Conference on*, vol. 0, pp. 672–681, 2005.
- [39] A. Hossain, P. Biswas, and S. Chakrabarti, “Sensing models and its impact on network coverage in wireless sensor network,” in *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference on*, Dec. 2008, pp. 1–5.
- [40] L. Lazos and R. Poovendran, “Coverage in heterogeneous sensor networks,” in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, April 2006, pp. 1–10.

- [41] S. Dhillon and K. Chakrabarty, “Sensor placement for effective coverage and surveillance in distributed sensor networks,” in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, March 2003, pp. 1609–1614 vol.3.
- [42] M. A. Batalin and G. S. Sukhatme, “Coverage, exploration and deployment by a mobile robot and communication network,” in *Proceedings of the International Workshop on Information Processing in Sensor Networks*, Palo Alto Research Center (PARC), Palo Alto, April 2003, pp. 376–391.
- [43] A. Kulakov and D. Davcev, “Distributed algorithm for a mobile wireless sensor network for optimal coverage of non-stationary signals,” in *Spatial Stochastic Modeling of Wireless Networks, 2005. SpaSWiN 2005*, April 2005.
- [44] P.-C. Wang, T.-W. Hou, and R.-H. Yan, “Maintaining coverage by progressive crystal-lattice permutation in mobile wireless sensor networks,” in *Systems and Networks Communications, 2006. ICSNC '06. International Conference on*, Oct. 2006, pp. 42–42.
- [45] P.-J. Wan and C.-W. Yi, “Coverage by randomly deployed wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2658–2669, 2006.
- [46] J. W. J.H. Ahlberg, E.N. Nilson, *The Theory of Splines and Their Applications*. Academic Press, 1967.
- [47] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [48] E. Candes and M. Wakin, “An introduction to compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, March 2008.

- [49] S. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 674–693, Jul 1989.
- [50] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” 2008. [Online]. Available: <http://www.citebase.org/abstract?id=oai:arXiv.org:0808.3572>
- [51] D. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [52] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [53] T. T. Emmanuel J. Cands, Justin K. Romberg, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [54] E. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007. [Online]. Available: <http://stacks.iop.org/0266-5611/23/969>
- [55] M. Mishali and Y. Eldar, “The continuous joint sparsity prior for sparse representations: Theory and applications,” in *Computational Advances in Multi-Sensor Adaptive Processing, 2007. CAMPSAP 2007. 2nd IEEE International Workshop on*, Dec. 2007, pp. 125–128.

- [56] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss, “Random sampling for analog-to-information conversion of wideband signals,” in *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, Oct. 2006, pp. 119–122.
- [57] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud, “Theory and implementation of an analog-to-information converter using random demodulation,” in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, May 2007, pp. 1959–1962.
- [58] I. Maravic and M. Vetterli, “Digital ds-cdma receivers working below the chip rate,” in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, vol. 2, Nov. 2002, pp. 1463–1467 vol.2.
- [59] H. Lee and Z. Bien, “Sub-nyquist nonuniform sampling and perfect reconstruction of speech signals,” in *TENCON 2005 2005 IEEE Region 10*, Nov. 2005, pp. 1–6.
- [60] I. Maravic, M. Vetterli, and K. Ramchandran, “Channel estimation and synchronization with sub-nyquist sampling and application to ultra-wideband systems,” in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 5, May 2004, pp. V-381–V-384 Vol.5.
- [61] I. Maravic and M. Vetterli, “Sampling and reconstruction of signals with finite rate of innovation in the presence of noise,” *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2788–2805, Aug. 2005.

- [62] M. Vetterli, P. Marziliano, and T. Blu, “Sampling signals with finite rate of innovation,” *Signal Processing, IEEE Transactions on*, vol. 50, no. 6, pp. 1417–1428, Jun 2002.
- [63] L. Berman and A. Feuer, “Robust patterns in recurrent sampling of multiband signals,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2326–2333, June 2008.
- [64] R. Venkataramani and Y. Bresler, “Optimal sub-nyquist nonuniform sampling and reconstruction for multiband signals,” *Signal Processing, IEEE Transactions on*, vol. 49, no. 10, pp. 2301–2313, Oct 2001.
- [65] H. Lee and Z. Bien, “Sub-nyquist nonuniform sampling and perfect reconstruction of signals,” in *Intelligent Signal Processing and Communication Systems, 2004. ISPACS 2004. Proceedings of 2004 International Symposium on*, Nov. 2004, pp. 293–298.
- [66] D. Needell and J. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301 – 321, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/B6WB3-4T1Y404-1/2/a3a764ae1efc1bd0569dcde301f0c6f1>
- [67] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, “Beyond nyquist: Efficient sampling of sparse bandlimited signals,” 2009. [Online]. Available: <http://arxiv.org/abs/0902.0026>
- [68] R. Coifman and M. Wickerhauser, “Entropy-based algorithms for best basis selection,” *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 713–718, Mar 1992.
- [69] J. Galambos, *The Asymptotic Theory of Extreme Order Statistics*, 2nd ed. Robert E. Krieger Publishing Company, 1987.

- [70] T. L. Tung, K. Yao, C. W. Reed, R. E. Hudson, D. Chen, and J. Chen, “Source localization and time delay estimation using constrained least squares and best path smoothing,” in *Proc. SPIE*, vol. 3807, 7 1999, pp. 220–231.
- [71] Wikipedia, “Order statistic — Wikipedia, the free encyclopedia,” 2007, [Online; accessed August-22-2007]. [Online]. Available: [http://en.wikipedia.org/wiki/Order\\\_statistic](http://en.wikipedia.org/wiki/Order\_statistic)
- [72] *Lowering Cost and Improving Interoperability by Predicting Residual BER: Theory, Measurements, and Applications*, Application Note 1397-1, Agilent, 2002.
- [73] B. Grocholsky, “Information-theoretic control of multiple sensor platforms,” Ph.D. dissertation, The University of Sydney, 2002.
- [74] L. Yip, J. Chen, R. Hudson, and K. Yao, “Cramér-rao bound analysis of wideband source localization and doa estimation,” in *Proceedings of SPIE Advance Signal Processing Algorithms, Architectures and Implementations XII*, vol. 4791, 2002, pp. 304–316.
- [75] R. D. Fierro and K. Yao, “Analytical bounds on least squares and total least squares methods for the linear prediction problem,” UCLA, Computational and Applied Mathematics (CAM) 93-10, May 1993.
- [76] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, “Parameter estimation in the presence of bounded data uncertainties,” *SIAM J. Matrix Anal. Appl.*, vol. 19, no. 1, pp. 235–252, 1998.

- [77] S. V. Huffel and J. Vandewalle, *The Total Least Squares Problem Computational Aspects and Analysis*. Philadelphia, PA: SIAM, 1991.
- [78] A. van der Sluis, “Stability of the solutions of linear least squares problems,” *Numerische Mathematik*, vol. 23, pp. 241–254, 1975.
- [79] A. van der Sluis and G. Veltkamp, “Restoring rank and consistency by orthogonal projection,” *Linear Algebra and its Applications*, vol. 28, pp. 257 – 278, 1979. [Online]. Available: <http://www.sciencedirect.com/science/article/B6V0R-45FKG05-32/2/e94ebd10533e8e5b2c2c671589d24da7>
- [80] D. Li, K. Wong, Y. Hu, and A. Sayeed, “Detection, classification, and tracking of targets,” *Proc. IEEE Signal Processing Magazine*, vol. 19, pp. 17–31, Mar. 2002. [Online]. Available: [citeseer.ist.psu.edu/li02detection.html](http://citeseer.ist.psu.edu/li02detection.html)
- [81] R. Brooks, P. Ramanathan, and A. Sayeed, “Distributed target classification and tracking in sensor networks,” *Proceeding IEEE*, vol. 91, no. 8, aug 2003.
- [82] A. Pagés-Zamora, J. Vidal, and D. H. Brooks, “Closed-form solution for positioning based on angle of arrival measurements,” in *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC’02)*, 2002.
- [83] D. Torrieri, “Statistical theory of passive location systems,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-20, no. 2, pp. 183–198, March 1984.
- [84] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.

- [85] D. Niculescu and B. Nath, “Ad hoc positioning system,” 2001. [Online]. Available: [citeseer.ist.psu.edu/niculescu01ad.html](http://citeseer.ist.psu.edu/niculescu01ad.html)
- [86] Y. Sung, L. Tong, and H. V. Poor, “Sensor configuration and activation for field detection in large sensor arrays,” 2005. [Online]. Available: <http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0502080>
- [87] N. Bulusu, J. Heidemann, and D. Estrin, “Adaptive beacon placement,” in *Distributed Computing Systems, 2001. 21st International Conference on.*, Apr. 2001, pp. 489–498. [Online]. Available: [citeseer.ist.psu.edu/article/bulusu01adaptive.html](http://citeseer.ist.psu.edu/article/bulusu01adaptive.html)
- [88] Y. Mostofi, T. H. Chung, R. M. Murray, and J. W. Burdick, “Communication and sensing trade-offs in decentralized mobile sensor networks: A cross-layer design approach,” in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, Apr. 2005, pp. 118–125.
- [89] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, “Convex position estimation in wireless sensor networks,” in *Proc. IEEE INFOCOM*, vol. 3, Anchorage, AK, Apr. 2001, pp. 1655–1663.
- [90] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, “Ecolocation: a sequence based technique for rf localization in wireless sensor networks,” in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 38.
- [91] Y.-C. Tong and G. Pottie, “The marginal utility of cooperation in sensor networks,”

- Information Theory and Applications Workshop, 2008*, pp. 256–262, 27 2008-Feb. 1 2008.
- [92] A. Krause and C. Guestrin, “Nonmyopic active learning of gaussian processes: an exploration-exploitation approach,” in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 449–456.
- [93] A. Fanimokun and J. Frolik, “Effects of natural propagation environments on wireless sensor network coverage area,” in *System Theory, 2003. Proceedings of the 35th Southeastern Symposium on*, March 2003, pp. 16–20.
- [94] J. Liu, X. Koutsoukos, J. Reich, and F. Zhao, “Sensing field: coverage characterization in distributed sensor networks,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 5, April 2003, pp. V–173–6 vol.5.
- [95] J. Adriaens, S. Megerian, and M. Potkonjak, “Optimal worst-case coverage of directional field-of-view sensor networks,” in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, Sept. 2006, pp. 336–345.
- [96] R. Venkataramani and Y. Bresler, “Perfect reconstruction formulas and bounds on aliasing error in sub-nyquist nonuniform sampling of multiband signals,” *Information Theory, IEEE Transactions on*, vol. 46, no. 6, pp. 2173–2183, Sep 2000.
- [97] G. Walter, “A sampling theorem for wavelet subspaces,” *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 881–884, Mar 1992.