UNIVERSITY OF CALIFORNIA
RIVERSIDE

Energy Efficient Distributed Data Fusion In Multihop Wireless Sensor Networks

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

Doctor of Philosophy

in

Electrical Engineering

by

Yi Huang

June 2010

Dissertation Committee:

Professor Yingbo Hua, Chairperson
Professor Ilya Dumer
Professor Sheldon Tan

The Dissertation of Yi Huang is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to take this opportunity to express my sincere gratitude to my advisor, Dr. Yingbo Hua, for his persistent support, continuous guidance, and invaluable advice during my Ph.D. study. Without his suggestion, encouragement, patience, and conscientiousness, I would not be able to successfully complete the dissertation. It has been a great pleasure to have him as my advisor.

I would also like to thank Dr. Ilya Dumer and Dr. Sheldon Tan for serving on my committee.

I would like to express my special thanks to Dr. Ananthram Swami from Army Research Laboratory for his support for this work.

I would express my appreciation to all the members in Laboratory of Signals, Systems and Networks, past and present, from whom I learned a great deal and with whom I shared my laughs. Thanks to Dr. Bin Zhao, Dr. Yue Rong for their helpful discussion. Thanks to Dr. Yan Mei, Dr. Zheng Fang, Dr. Kezhu Hong, Dr. Yuan Yu, Ting Kong, Jie Liang, Shengyang Xu, Xiang Dong, Haitao Liu, Qian Gao, and Ali Cirik for their help and friendship.

Last but not least, I would like to express my deepest gratitude to my family. They provide a constant source of motivation in my life. It is to them that I dedicate this work.

To Li and Christina.

ABSTRACT OF THE DISSERTATION

Energy Efficient Distributed Data Fusion In Multihop Wireless Sensor Networks

by

Yi Huang

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2010
Professor Yingbo Hua, Chairperson

This thesis addresses a transmission energy problem for wireless sensor networks. There are two types of wireless sensor networks. One is single-hop sensor network where data from each sensor is directly transmitted to a fusion center, and the other is multihop sensor network where data is relayed through adjacent sensors. In the absence of a moving agent for data collection, multihop sensor network is typically much more energy efficient than single-hop sensor network since the former avoids long distance data transmission. Progressive data fusion is a distributed fusion method that fuses data as they hop through sensors, which is effective to further reduce the energy cost. With the knowledge of a routing tree and all channel state information, the transmission energy allocated for each sensor can be pre-determined to even further reduce the energy cost while satisfying a pre-determined performance. In this thesis, we develop several energy planning algorithms for the above purpose. Specifically we designed two energy planning algorithms for progressive estimation with digital transmissions between sensors and one energy planning algorithm for progressive estimation with analog transmission. We also show that digital transmission is

more efficient in transmission energy than analog transmission if the available transmission time-bandwidth product for each link and each observation sample is not too limited.

We also study energy cost for consensus estimation which is a distributed fusion method for peer-to-peer multihop sensor networks. The impact of fusion weights and energy allocation for each sensor is also investigated. We demonstrate that to achieve an approximately same performance, the total energy cost for consensus estimation can be much higher than that for progressive estimation, but the peak energy for the former is less than that for the latter.

Contents

List of Figures **x**

List of Tables **xiii**

1 Introduction **1**
1.1 Wireless Sensor Networks 1
1.2 Distributed Data Fusion 2
1.3 Energy Planning 3
1.4 Multihop versus single-hop 5
1.5 Thesis Overview 6

2 Data Fusion Frameworks **9**
2.1 Progressive Data Fusion 9
2.1.1 Progressive Estimation 9
2.1.2 Progressive Detection 14
2.2 Consensus Data Fusion 19
2.3 Single-hop Data Fusion 22
2.4 Summary 25

3 Energy Planning for Progressive Data Fusion **26**
3.1 System Model 26
3.2 Communication Channel Model 28
3.3 Probabilistic Uniform Quantization 31
3.4 Analog Transmission versus Digital Transmission 32
3.5 Progressive Estimation with Digital Transmission 34
3.5.1 BLUE 34
3.5.2 Quasi BLUE 36
3.5.3 Averaging 36
3.6 Energy Planning with Digital Transmission 37
3.6.1 Algorithm 1 37
3.6.2 Algorithm 2 43

3.6.3 Linear Energy Model for Progressive Estimation 49
3.7 Progressive Estimation with Analog Transmission 51
3.7.1 BLUE 52
3.7.2 Averaging 53
3.8 Energy Planning with Analog Transmission 53
3.9 Simulation Results 56
3.9.1 Analog Transmission Energy Planning 57
3.9.2 Digital Transmission Energy Planning - Algorithm 1 62
3.9.3 Digital Transmission Energy Planning - Algorithm 2 64
3.9.4 Analog versus Digital transmissions 69
3.9.5 Progressive versus Single-hop on Energy Cost 71
3.9.6 Robustness 72
3.10 Summary 72
3.11 Appendix 74
3.11.1 Pulse Modulation 74
3.11.2 Proof of Lemma 1 75

4 Energy Planning for Consensus Data Fusion **81**
4.1 System Model 81
4.2 Communication Protocol 83
4.3 Collision Avoidance 86
4.4 Fusion Algorithms 91
4.5 Energy Planning 93
4.5.1 Problem Formulation and Simplification 93
4.5.2 Optimal Selection of γ 96
4.5.3 Other Selections of γ 98
4.6 Simulation Results 99
4.6.1 Effect of γ selection 100
4.6.2 Effect of E_0 and B 100
4.6.3 Consensus versus Progressive on Energy Cost 101
4.7 Summary 105

5 Application Examples **108**
5.1 Camera Network Assisted Football Player Movement Tracking 108
5.2 Sensor Network Assisted Targets Tracking 111

6 Conclusion **115**

Bibliography **118**

List of Figures

2.1 Portion of a routing path where each sensor has only one upper stream sensor. 10
2.2 Portion of a routing path where each sensor may have more than one upper stream sensors. 11
2.3 The probability of the progressive detection error $p_{e,k}$ versus the sensor index k. The numbers of bits for quantization shown in this plot are used to quantize $p_{e,k-1}$ at sensor $k-1$ nonlinearly. The other parameters used for the quantization are $p_e^* = 0.04$ and $\epsilon = 0.19$. 19
2.4 The demonstration of the local fusion rule for the consensus based estimation. 21
2.5 The single hop data fusion system model with distributed data observation and processing for a wireless sensor networks. Each circle 'S' stands for one sensor and the box 'P' is the local data processor. 23

3.1 The snapshot of a network tree applying the progressive estimation. The solid biggest node in the middle is the destination node. 28
3.2 A 2-D multihop sensor network with a routing tree. The destination node (also referred to as the fusion center) is marked by the circle in the center. Here, there are 400 sensors, each marked by $*$. This network is used for all simulation examples in this thesis. 59
3.3 For the analog PP and UP algorithms, the transmission energy consumed by the network versus the target MSE_0. Note that the first curve on the top is based on a single-hop tree for the same distribution of sensors shown in Fig. 3.2. The path loss exponent is assumed to be four. 60
3.4 For the analog PP and UP algorithms, the transmission energy consumed by each sensor versus the normalized Euclidean distance between the sensor and the fusion center. $MSE_0 = 0.05$. The analog PP algorithm uses F_1. . . 60
3.5 Illustration of the effect of p in F_p for the analog PP algorithm. 61
3.6 For the analog PP and UP algorithms, the actual MSE at the fusion center versus the target MSE. 61
3.7 For the digital PP and UP algorithms, the transmission energy consumed by the network versus the target MSE where p is as in J_p for the digital PP algorithm. 65

3.8 The number of quantization bits allocated for each sensor per element of the unknown vector versus the normalized Euclidean distance from the sensor to the destination node. The target MSE at the destination node is $MSE_0 = 1.0 \times 10^{-3}$. The digital PP algorithm uses J_1. 65
3.9 Normalized digital transmission energy by each sensor versus the normalized Euclidean distance from the sensor to the destination node. $MSE_0 = 1.0 \times 10^{-3}$. The digital PP algorithm uses J_1. 66
3.10 Same as Figure 3.8 except $p = 50$ for the digital PP algorithm. 66
3.11 Same as Figure 3.9 except $p = 50$ for the digital PP algorithm. 67
3.12 Total digital transmission energy consumed by the network versus L/M with $M = 10$. $MSE_0 = 1.0 \times 10^{-3}$. 67
3.13 For the digital PP and UP algorithms, the actual MSE value at the destination node versus the target MSE MSE_0. 70
3.14 The sum energy required by different energy planning algorithms for progressive estimation. 70
3.15 The sum energy required by four different energy planning algorithms for progressive estimation versus the time-bandwidth product L, where $MSE_0 = 0.01$. For digital PP algorithm 1 and 2, $p = 1$ was used. 77
3.16 Amount of transmission energy consumed by an individual sensor versus the sensor index for the digital PP algorithm 2 for progressive estimation, with $p = 1$, 8 and 64, where $MSE_0 = 0.001$. The sensor index is sorted increasingly as the distance between the sensor and the fusion center increases. For progressive estimation, sensor zero in this and the following figures is the fusion center. 78
3.17 Comparison between analog transmissions and digital transmissions: total transmission energy versus the target MSE. 79
3.18 Total amount of normalized transmission energy consumed by the network versus MSE_0. The network used is Fig. 3.2. 79
3.19 Digital transmissions: Actual MSE versus node failure rate. The target MSE is $MSE_0 = 0.002$. 80
3.20 Analog transmissions: Actual MSE versus node failure rate. The target MSE is $MSE_0 = 0.002$. 80

4.1 Frame structure of the distributed SAM protocol, which resembles that of MSH-DSCH in IEEE 802.16. 86
4.2 A snapshot of concurrent co-channel transmissions determined by the D-SAM protocol for a network in a regular square grid. $\mathscr{R} = d_a$, $M = 8$, where d_a is the minimum distance between two adjacent nodes. The black nodes are the receiving nodes, and the grey nodes are the transmitting nodes. 90
4.3 $MSE_1(t, \gamma)/K$ of consensus estimation under (4.5) versus the iteration index t, with different γ values: $\gamma_1 = 0.0771$ (optimized with quantization errors), $\gamma_2 = 0.2163$ (optimized without quantization errors) and $\gamma_3 = 0.1429$ (using maximum degree of connectivity), where $E_0 = 1280$, $B = 7$, $MSE_0 = 0.001$. 101

4.4 The contour of the sum energy required by consensus estimation under (4.5) in terms of E_0 and B. The target MSE is set at $MSE_0 = 0.001$. 102
4.5 The optimal B, E_0 and t required by consensus estimation versus MSE_0. . 103
4.6 Sum energy cost for consensus and progressive estimation. 104
4.7 Amount of transmission energy (in log scale) consumed by an individual sensor versus the sensor index, where $MSE_0 = 0.001$. For the digital PP algorithm 2, the curve for $p = 1$ is shown here, and the curves for other values of p would differ slightly due to the log scale. 106
4.8 Averaged number of quantization bits per element allocated by the digital PP algorithm 2, the digital UP algorithm, and the consensus algorithm, for progressive estimation for an individual sensor versus the sensor index, where $MSE_0 = 0.001$. 107

5.1 The cameras deployment diagram for the football field. The blank little circles are cameras, and the solid big circles denotes the fusion center. . . . 109
5.2 The actual and tracked offense players movements in one possession. 111
5.3 The actual target trajectory and the estimated trajectory. 114

List of Tables

4.1 Total transmission energy consumed by the network under different choices of E_0 and B. The MSE target is set $MSE_0 = 0.001$. 105

Chapter 1

Introduction

1.1 Wireless Sensor Networks

Recent advances in micro-electronic and mechanical technologies have enabled us to incorporate the operational units of sensing, data processing, and wireless communication into a small size device. Such a device can be called a sensor for convenience. The sensing unit collects information from the surrounding environment. The processing unit performs some local information processing, such as quantization and compression. The wireless communications unit allows every sensor to share information with each other for better data fusion intelligence.

A wireless sensor network may consist of hundreds, or even thousands of such inexpensive sensors spread across a geographical area. When properly configured and deployed, such networked sensors can collaborate to accomplish a wide range of tasks such as battlefield surveillance, homeland security, environmental monitoring, emergency response,

traffic control, healthcare, and home automation, just to name a few.

For example, in homeland security, solar powered bio-sensors can be deployed along national borders to detect the smuggling of bio-weapons by terrorists, and networked video, acoustic, or other types of sensors can be used to track suspected targets. In environmental monitoring, sensor networks can be deployed to monitor the natural environment and wild animals' activities. In emergency response, a large number of sensors can be dropped in to assist rescue operations by locating survivors or identifying risky areas. Image sensors and other types of sensors have been used at roadway intersections to monitor traffic conditions or identify vehicles. In the healthcare industry, the use of body-area sensor networks is useful for monitoring the vital signs of patients. Interested readers can refer to [1] for more potential applications.

1.2 Distributed Data Fusion

Data fusion is the fundamental to support the aforementioned applications in sensor networks. Data fusion in sensor networks is generally defined as combining noise corrupted data from multiple sensors to acquire the desired information at a fusion center. In other words, data are observed at sensors, and then transported to and fused at the fusion center.

Since data collected by sensors are distributed at different geographical locations, data fusion using a wireless sensor network requires not only local information processing but also inter-sensor communication. The latter adds wireless communication and networking into the problem that is absent from the traditional centralized data fusion where all data

are available at the fusion center. Specifically, there are three challenges with distributed data fusion in wireless sensor networks, which are listed below.

1) Energy Consumption: Energy consumption is a major concern of wireless sensor networks, since sensors are only equipped with limited batteries. Furthermore, the replacement of batteries can be costly, even impossible for some applications. Therefore, sensor network operations must be developed in an energy-efficient manner for network lifetime maximization and performance stability. One major objective of this research is to invent new energy planning schemes for data fusion in wireless sensor networks.

2) Large Network Size: A wireless sensor network has a large number (could be thousands) of cheap and low-power sensors deployed in a certain geographic area. On one hand, this has a great potential to improve the fusion precision and network robustness. But on the other hand, this creates problems for sensor collaboration and networking, which require distributed and scalable solutions for data fusion.

3) Self-Configuration: In a large sensor network, there could be little or no pre-established infrastructure. Furthermore, the network topology may change dynamically because the nodes may sleep, fail, or move. Wireless sensor networks are required to have the capability of self-configuration. This includes, for example, sensor self-localization, coverage control, and adaptation to communication failure in a harsh wireless environment.

1.3 Energy Planning

As we discussed in Section (1.2), energy is a big concern for data fusion in sensor networks. Various energy-efficient algorithms have been proposed for network coverage [2],

data gathering [3], and protocols of medium access control [4] and routing [5].

This thesis will focus on designing energy planning algorithms to achieve energy efficient data fusion in wireless sensor networks. By definition, energy planning is to use the least amount of energy to perform data fusion in the sensor network, while still guarantee the data fusion performance.

There are two parts of energy consumption in sensor networks. One is for observation and on-chip signal processing, which is basically the energy used on circuit board inside each sensor. With the advances of circuit technology, this part of energy can be decreased gradually. The more dominating part of energy consumption is actually on wireless communications between sensors for information sharing. This energy is basically governed by the physical laws, which are consistent over time. If not carefully planned, this part of energy would be overwhelming and drain the batteries quickly. For convenience, hereafter in this paper, we use energy to refer communication energy. Observation and signal processing energy planning is out of the scope of this thesis.

Several other assumptions need to be made for the (communication) energy planning problem in sensor networks. First, we assume a central planner has all the knowledge of the network including the network topology, wireless channel state information between any two nodes, and a pre-given data fusion target performance. Based on these knowledge, the planner will design the best energy planning for the the network to use least amount of energy to achieve the target performance. After the energy planning is designed, it will be applied to the network and distributed data fusion will be conducted over the network according to the planning. Second, the network is also assumed to be stationary during the

running period of a designed energy planning. Once the network conditions are changed, say new sensors joining in the network, existing sensors failure, or wireless channel varies among sensor, the energy planning need to be re-designed to adapt to those changes.

1.4 Multihop versus single-hop

A sensor network can be a single-hop sensor network or a multihop sensor network. There also appear three basic architectures for wireless sensor networks: single-hop network with fusion center, multihop network with fusion center, and peer-to-peer multihop network. Clearly, many other architectures can be formed by combinations of the three.

A single-hop network with fusion center is a conventional sensor network and, if affordable, is often the most effective for the fusion performance. An example of such a network comprises sensors on the ground and a flying-over vehicle collecting data from each sensor, where a sensor transmits data only when the vehicle becomes close enough. For reducing the energy cost in single-hop sensor networks, there have been many research activities as reported in [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] and the references therein. However, for typical flat networks in outdoor environment or 3-D networks inside large buildings, direct communications between each sensor and a fixed fusion center would make the network very inefficient in energy cost because of transmission path losses.

A multihop network with fusion center is suitable for the above mentioned situation where data packets hop from one sensor to another until they reach the fusion center. However, if the data packets are not compressed as they hop towards the fusion center,

the sensors near the fusion center can be over burdened - a bottleneck effect. To solve this problem, progressive estimation [8, 23] is a useful idea where data are fused together as they hop from one sensor to another along a routing path. To further reduce the energy cost at all sensors, the design of the number of bits for quantization at each sensor is also useful as shown in [24, 25].

A peer-to-peer multihop network is useful for situations where there is no centralized control or the network is too dynamic to be regulated centrally. There are numerous articles in this area. A major approach for distributed fusion in peer-to-peer multihop network is known as consensus estimation [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45] where sensors exchange information with their neighbors and iteratively update their own information. Although consensus estimation does not require centralized control, their convergence to a desired result still depends on the network topology. Without the knowledge of the network topology, a node in the network may not even know when and how an updated information is reliable enough.

1.5 Thesis Overview

In this dissertation, we examine the distributed data fusion problem in multihop wireless sensor networks in which the main design goal are energy efficiency. Specifically, the dissertation is organized as follows.

Chapter 2 introduces three main categories of distributed data fusion frameworks in sensor networks. Namely, they are progressive data fusion, consensus data fusion and single-hop data fusion. Progressive data fusion is the original contribution of this disserta-

tion, while the other two are pre-existing in the literature. In Section 2.1.1, we present the progressive estimation algorithms based on BLUE and LMMSE, where an interesting similarity will be observed. In Section 2.1.2, we present the progressive detection algorithms using either estimation propagation or decision propagation, the latter of which further reduces the communication burden.

Chapter 3 focuses on energy planning for progressive data fusion. We first describe the model of the data observed by each sensor, the communication channel model and the energy model for data transmission among sensors, and a probabilistic uniform quantization scheme. We then compares the errors caused by analog and digital communications. Section 3.5 presents several progressive estimation algorithms with digital transmissions. These algorithms are based on the best linear unbiased estimation (BLUE), quasi BLUE and a simple averaging. The recursive inequality of the covariance matrices based on the averaging algorithm is used for the design of the digital transmission energy planning to be shown in section 3.6. We present two algorithms for energy planning with digital transmissions. In section 3.7, progressive estimation for multihop networks with analog transmissions are proposed, which are based on BLUE and another simple averaging. The analog transmission energy planning is designed in section 3.8. The simulation examples are given in section 3.9.

Chapter 4 focuses on energy planning for consensus data fusion. we first propose the system model and the communication protocol for consensus estimation. Many fundamental issues such as transmission energy, number of quantization bits, and the condition for successful packet reception are considered. We then present a realistic media access

control (MAC) protocol called distributed array method (D-SAM) to avoid packet collision for communications with this consensus data fusion. We then consider applying three different fusion algorithms for the consensus estimation. To achieve a better understanding of the energy cost for consensus estimation, we also present an energy planning algorithm for consensus estimation. Fianlly, we present simulation examples to compare the performance of the new energy planning algorithm for progressive estimation with those of several other alternatives. We illustrate the effects of several fusion weight choices, quantization bits allocation, and energy allocation for each sensor on the performance of consensus estimation in the presence of quantization errors. We also compare the minimized energy for consensus estimation with that for progressive estimation.

Chapter 5 describe two realistic applications of progressive estimation using wireless sensor networks. One application is to use a camera network to track the movement of the players simultaneously in a field game. Another application is to use a sensor network to monitor and track multiple aircrafts in region of interest.

Chapter 6 concludes the main original contributions of this dissertation.

Chapter 2

Data Fusion Frameworks

2.1 Progressive Data Fusion

2.1.1 Progressive Estimation

Consider the following model for the data observed at sensor k:

$$\boldsymbol{x}_k = \boldsymbol{G}_k\boldsymbol{\theta} + \boldsymbol{n}_k \tag{2.1}$$

where $\boldsymbol{x}_k$ is the observation data vector, $\boldsymbol{n}_k$ is the observation noise vector of zero mean, $\boldsymbol{G}_k$ is the observation matrix of full column rank, and $\boldsymbol{\theta}$ is an unknown vector parameter invariant to the space (or sensor) index k. The observation matrix $\boldsymbol{G}_k$ depends on the characteristics of sensor k and its location with respect to the unknown vector $\boldsymbol{\theta}$. Since $\boldsymbol{G}_k$ is arbitrary, a simple normalization of (2.1) can ensure that the covariance matrix of each noise vector is the identify matrix $\boldsymbol{I}$.

We next show two algorithms for progressive estimation: one is based on the best (minimum variance) linear unbiased estimation (BLUE) and the other is based on the linear

minimum mean square error (LMMSE) estimation.

BLUE

We denote an unbiased estimator of $\boldsymbol{\theta}$ at sensor k by $\hat{\boldsymbol{\theta}}_k$ and its covariance matrix by $\boldsymbol{C}_k$. We let sensor $k-1$ be the upper stream sensor to sensor k as illustrated in Figure 2.1. We assume that sensor $k-1$ transmits $\hat{\boldsymbol{\theta}}_{k-1}$ and $\boldsymbol{C}_{k-1}$ to sensor k before sensor k computes $\hat{\boldsymbol{\theta}}_k$ and $\boldsymbol{C}_k$.

Figure 2.1: Portion of a routing path where each sensor has only one upper stream sensor.

Assume that sensor 1 does not have any upper stream sensor, then the BLUE estimate of $\boldsymbol{\theta}$ at sensor 1 is given by

$$\hat{\boldsymbol{\theta}}_1 = \left(\boldsymbol{G}_1^H \boldsymbol{G}_1\right)^{-1} \boldsymbol{G}_1^H \boldsymbol{x}_1 \tag{2.2}$$

and its covariance matrix is

$$\boldsymbol{C}_1 = \left(\boldsymbol{G}_1^H \boldsymbol{G}_1\right)^{-1} \tag{2.3}$$

Provided that $\boldsymbol{x}_k$ and $\hat{\boldsymbol{\theta}}_{k-1}$ are uncorrelated (which is guaranteed if the noises $\boldsymbol{n}_k$ and $\boldsymbol{n}_l$ are uncorrelated for $k \neq l$), the BLUE estimate of $\boldsymbol{\theta}$ at sensor k based on $\boldsymbol{x}_k$ and $\hat{\boldsymbol{\theta}}_{k-1}$ is given by

$$\hat{\boldsymbol{\theta}}_k = \left(\boldsymbol{C}_{k-1}^{-1} + \boldsymbol{G}_k^H \boldsymbol{G}_k\right)^{-1} \left(\boldsymbol{C}_{k-1}^{-1} \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{G}_k^H \boldsymbol{x}_k\right) \tag{2.4}$$

and the inverse of its covariance matrix is

$$\boldsymbol{C}_k^{-1} = \boldsymbol{C}_{k-1}^{-1} + \boldsymbol{G}_k^H \boldsymbol{G}_k \tag{2.5}$$

We see that $\boldsymbol{C}_k^{-1}$ is monotonically increasing with k and hence $\boldsymbol{C}_k$ is monotonically decreasing to zero as k increases. We also see that the update of $\boldsymbol{C}_k^{-1}$ is much simpler than the update of $\boldsymbol{C}_k$, which suggests that sensor $k-1$ should transmit $\boldsymbol{C}_{k-1}^{-1}$ (instead of $\boldsymbol{C}_{k-1}$) to sensor k. More importantly, the covariance matrix of the progressive estimation error at each sensor k is the same as that if sensor k performs the centralized estimation using all observations from its upper stream sensors. For sensor k to collect all observations $\{x_k\}$ from sensors 1 to $k-1$, there need to be $\sum_{i=1}^{k-1} = \frac{k(k-1)}{2} = O(k^2)$ transmissions in the network, which is in contrast to $O(k)$ transmissions required by the progressive estimation. Furthermore, the centralized estimator also needs to know all the associated observation matrices $\{\boldsymbol{G}_k\}$. Clearly, if a large number of sensors is needed to meet a required estimation accuracy, the progressive estimation is more desirable than the centralized estimation. It should be noted that at each hop of the progressive estimation, either $\boldsymbol{C}_{k-1}$ or $\boldsymbol{C}_{k-1}^{-1}$ needs to be transmitted along with a local estimate $\hat{\boldsymbol{\theta}}_{k-1}$ of the unknown vector $\boldsymbol{\theta}$. If the dimension of the unknown vector $\boldsymbol{\theta} \in C^{N\times 1}$ is so large that N^2 is comparable to K^2 where K is the "last" sensor, the advantage of the progressive estimation may not be realized.

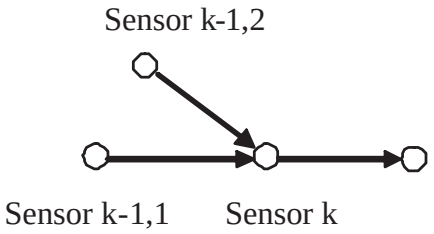

Figure 2.2: Portion of a routing path where each sensor may have more than one upper stream sensors.

More generally, as illustrated in Figure 2.2, if sensor k receives from its upper

stream sensors more than one uncorrelated estimates of $\boldsymbol{\theta}$, denoted by $\hat{\boldsymbol{\theta}}_{k-1,i}$ with the corresponding covariance matrices $\boldsymbol{C}_{k-1,i}$ where $i = 1, ..., I_k$, then the BLUE estimate of $\boldsymbol{\theta}$ at sensor k is given by

$$\hat{\boldsymbol{\theta}}_k = \left(\sum_{i=1}^{I_k} \boldsymbol{C}_{k-1,i}^{-1} + \boldsymbol{G}_k^H \boldsymbol{G}_k \right)^{-1} \left(\sum_{i=1}^{I_k} \boldsymbol{C}_{k-1,i}^{-1} \hat{\boldsymbol{\theta}}_{k-1,i} + \boldsymbol{G}_k^H \boldsymbol{x}_k \right) \tag{2.6}$$

and the corresponding covariance matrix is

$$\boldsymbol{C}_k = \left(\sum_{i=1}^{I_k} \boldsymbol{C}_{k-1,i}^{-1} + \boldsymbol{G}_k^H \boldsymbol{G}_k \right)^{-1} \tag{2.7}$$

The above progressive estimation algorithm is distributed. Each sensor uses its locally available information. When the estimation accuracy at a sensor meets a pre-specified requirement, e.g., $tr(\boldsymbol{C}_k)$ becomes small enough, this sensor can stop the process of the progressive estimation and possibly send a message to trigger a network-wide operation.

LMMSE

The previous BLUE estimation treats $\boldsymbol{\theta}$ as an unknown deterministic vector. For the LMMSE estimation, we will treat $\boldsymbol{\theta}$ as an unknown random vector with zero mean and a known covariance matrix. Since $\boldsymbol{G}_k$ is arbitrary, a simple normalization of (2.1) can ensure that the covariance matrix of $\boldsymbol{\theta}$ and the covariance matrix of the noise $\boldsymbol{n}_k$ are both equal to the identity matrix $\boldsymbol{I}$.

At sensor 1, the LMMSE estimate of $\boldsymbol{\theta}$ is given by

$$\tilde{\boldsymbol{\theta}}_1 = \boldsymbol{G}_1^H \left(\boldsymbol{G}_1 \boldsymbol{G}_1^H + \boldsymbol{I} \right)^{-1} \boldsymbol{x}_1 \tag{2.8}$$

and its mean square error (MSE) matrix is

$$\boldsymbol{M}_1 = \boldsymbol{I} - \boldsymbol{G}_1^H \left(\boldsymbol{G}_1 \boldsymbol{G}_1^H + \boldsymbol{I} \right)^{-1} \boldsymbol{G}_1 \tag{2.9}$$

Assume that sensor k receives the LMMSE estimate $\tilde{\boldsymbol{\theta}}_{k-1}$ and the MSE matrix $\boldsymbol{M}_{k-1}$ from sensor $k-1$. Then, the LMMSE estimate of $\boldsymbol{\theta}$ at sensor k based on $\tilde{\boldsymbol{\theta}}_{k-1}$ and $\boldsymbol{x}_k$ is given by

$$\tilde{\boldsymbol{\theta}}_k = \boldsymbol{C}_{u,y_k}\boldsymbol{C}_{y_k,y_k}^{-1}\boldsymbol{y}_k \tag{2.10}$$

and its MSE matrix is

$$\boldsymbol{M}_k = \boldsymbol{I} - \boldsymbol{C}_{u,y_k}\boldsymbol{C}_{y_k,y_k}^{-1}\boldsymbol{C}_{u,y_k}^H \tag{2.11}$$

where

$$\boldsymbol{y}_k = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{k-1}^T & \boldsymbol{x}_k^T \end{bmatrix}^T \tag{2.12}$$

$$\boldsymbol{C}_{u,y_k} = \begin{bmatrix} \boldsymbol{I} - \boldsymbol{M}_{k-1} & \boldsymbol{G}_k^H \end{bmatrix} \tag{2.13}$$

$$\boldsymbol{C}_{y_k,y_k} = \begin{bmatrix} \boldsymbol{I} - \boldsymbol{M}_{k-1} & (\boldsymbol{I} - \boldsymbol{M}_{k-1})\boldsymbol{G}_k^H \\ \boldsymbol{G}_k(\boldsymbol{I} - \boldsymbol{M}_{k-1}) & \boldsymbol{G}_k\boldsymbol{G}_k^H + \boldsymbol{I} \end{bmatrix} \tag{2.14}$$

We see that like the BLUE estimation, the LMMSE estimation can also be formulated into a progressive form. Furthermore, one can verify by using (2.11) that

$$\boldsymbol{M}_k^{-1} = \boldsymbol{I} + \boldsymbol{C}_{u,y_k}\left(\boldsymbol{C}_{y_k,y_k} - \boldsymbol{C}_{u,y_k}^H\boldsymbol{C}_{u,y_k}\right)^{-1}\boldsymbol{C}_{u,y_k}^H \tag{2.15}$$

and by using (2.9), (2.13) and (2.14) that

$$\boldsymbol{M}_1^{-1} = \boldsymbol{I} + \boldsymbol{G}_k^H\boldsymbol{G}_k \tag{2.16}$$

$$\boldsymbol{M}_k^{-1} = \boldsymbol{M}_{k-1}^{-1} + \boldsymbol{G}_k^H\boldsymbol{G}_k \tag{2.17}$$

It is useful to notice the similarity between (2.17) and (2.5).

If the covariance matrix $\boldsymbol{C}_{u,u}$ of $\boldsymbol{\theta}$ is arbitrary but nonsingular, we can show that (2.17) still holds but (2.16) needs to be replaced by

$$\boldsymbol{M}_1^{-1} = \boldsymbol{C}_{u,u}^{-1} + \boldsymbol{G}_k^H \boldsymbol{G}_k \tag{2.18}$$

In general, we have that $\boldsymbol{M}_k < \boldsymbol{C}_k$ for all finite k, and $\boldsymbol{M}_k \to \boldsymbol{C}_k$ as k becomes large. But if $\boldsymbol{C}_{u,u}$ is so large that $\boldsymbol{C}_{u,u}^{-1}$ becomes negligible from (2.18), then $\boldsymbol{M}_k = \boldsymbol{C}_k$ for all k, i.e., the performance of BLUE becomes identical to that of LMMSE. In other words, an unknown deterministic vector is equivalent to a random vector of a covariance matrix with all eigenvalues infinitely large.

However, if sensor k receives multiple LMMSE estimates $\tilde{\boldsymbol{\theta}}_{k-1,i}$ and multiple MSE matrices $\boldsymbol{M}_{k-1,i}$ where $i = 1, ..., I_k$ from its upper stream sensors (see Figure 2.2), it does not appear possible in general to obtain the LMMSE estimate of $\boldsymbol{\theta}$ from $\boldsymbol{x}_k$, $\boldsymbol{G}_k$, $\tilde{\boldsymbol{\theta}}_{k-1,i}$ and $\boldsymbol{M}_{k-1,i}$, $i = 1, ..., I_k$. The problem lies with the fact that sensor k does not have sufficient information to compute such a cross-correlation matrix $\boldsymbol{C}_{\tilde{u}_{k-1,i},\tilde{u}_{k-1,j}}$ for $i \neq j$. This is a disadvantage of the LMMSE estimation.

2.1.2 Progressive Detection

At each sensor, the estimate of $\boldsymbol{\theta}$ obtained by progressive estimation can be used for progressive detection if multiple hypotheses are associated with different regions of $\boldsymbol{\theta}$. The BLUE estimation does not need any prior knowledge about $\boldsymbol{\theta}$ while the LMMSE estimation needs to know the covariance matrix of $\boldsymbol{\theta}$.

To understand the performance of the progressive detection, we now consider a

real-valued scalar data model for the observation at each sensor:

$$x_k = g_k\theta + n_k \tag{2.19}$$

which is simply a scalar version of (2.1) and the noise n_k is Gausssian with zero mean and unit variance. Furthermore, we treat a binary hypothesis problem where

$$\theta = \begin{cases} 0 & H_0 \\ a & H_1 \end{cases} \tag{2.20}$$

and $Pr(H_0) = Pr(H_1) = 0.5$.

Using estimation propagation

If the BLUE estimator $\hat{\theta}_k$ with the variance c_k is used at sensor k for detection, the detector with the minimum probability of error is: Decide H_1 if $\hat{\theta}_k > a/2$, or H_0 otherwise. The corresponding probability of error at sensor k is

$$p_{e,k} = Q\left(\frac{a}{2\sqrt{c_k}}\right) \tag{2.21}$$

where $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx$.

If the LMMSE estimator $\tilde{u}_k$ with the MSE m_k is used at sensor k for detection, the decision rule is the same: Decide H_1 if $\tilde{u}_k > a/2$, or H_0 otherwise. The detection error probability is given by (2.21) with c_k replaced by m_k. Note that for u defined in (2.20), the variance of u is $\frac{a^2}{2}$.

Using decision propagation

For both BLUE and LMMSE shown above, sensor k needs to receive two real numbers from sensor $k-1$. To reduce the communication burden, let us consider "decision

propagation" where sensor k receives the decision d_{k-1} (a binary number) and the corresponding probability of error $p_{e,k-1}$ (a real number) from sensor $k-1$. The detection at sensor k is solely based on d_{k-1}, $p_{e,k-1}$, x_k and g_k. Define

$$\lambda_k = \frac{P(d_{k-1}|H_1)f(x_k|H_1)}{P(d_{k-1}|H_0)f(x_k|H_0)} \tag{2.22}$$

where $P(\cdot|\cdot)$ denotes conditional probability and $f(\cdot|\cdot)$ denotes conditional probability density function. The optimal decision rule is: Decide H_1 if $\lambda_k > 1$, or H_0 otherwise.

The above decision rule can be simplified into:

- Decide H_1 if $d_{k-1} = 1$ and $x_k > \frac{a}{2} - \delta_k$ or if $d_{k-1} = 0$ and $x_k > \frac{a}{2} + \delta_k$;
- Decide H_0 if $d_{k-1} = 1$ and $x_k < \frac{a}{2} - \delta_k$ or if $d_{k-1} = 0$ and $x_k < \frac{a}{2} + \delta_k$.

Here, the offset δ_k is given by

$$\delta_k = \frac{1}{a} \ln \left(\frac{1 - p_{e,k-1}}{p_{e,k-1}} \right) \tag{2.23}$$

which is positive as long as $p_{e,k-1} < 0.5$. A quick observation from the above rule is that a higher hurdle is required if the decision at sensor k differs from that at sensor $k-1$, and a lower hurdle is required if the decision at sensor k coincides with that at sensor $k-1$. This extra hurdle is determined by δ_k.

The corresponding probability of error at sensor k is

$$p_{e,k} = Q\left(\frac{a}{2} - \delta_k\right) p_{e,k-1} + Q\left(\frac{a}{2} + \delta_k\right)(1 - p_{e,k-1}) \tag{2.24}$$

It is not yet proven that $p_{e,k}$ converges to zero as k increases. But numerical examples suggest that such a convergence property holds, which is illustrated later. Intuitively, we

see that when $p_{e,k-1}$ becomes smaller and hence δ_k becomes larger, $p_{e,k}$ is dominated by the first term in (2.24) and the ratio of $p_{e,k}$ over $p_{e,k-1}$ is dominated by $Q\left(\frac{a}{2}-\delta_k\right)$ which becomes closer to one.

A further reduction of the communication burden is possible if we use a finite number B of bits to quantize $p_{e,k-1}$ nonlinearly for each k. Specifically, we use p_e^* to denote a desired value of the probability of error, and ϵ for a small number such that $p_{e,1} < 0.5-\epsilon$. We then choose Δ such that $\Delta 2^B = \log_{10}(0.5-\epsilon) - \log_{10} p_e^*$. In other words, the number of bits required in terms of p_e^* is given by

$$B = \log_2 \left(\frac{1}{\Delta} \log_{10} \frac{0.5-\epsilon}{p_e^*} \right) \tag{2.25}$$

If the conventional linear quantization is used for $p_{e,k-1}$ for each k, the number of bits required would be $\log_2\left(\frac{1}{p_e^*}\right)$. At sensor $k-1$, an integer l_{k-1} is chosen such that $\log_{10} p_{e,k-1} \leq \Delta l_{k-1} + \log_{10} p_e^* \doteq \log_{10} \hat{p}_{e,k-1}$. Sensor k only receives d_{k-1} and l_{k-1} from sensor $k-1$, and reconstructs $\hat{p}_{e,k-1}$ from the above formula.

In order for the above strategy to work, we need to show that with $\hat{p}_{e,k-1}$, sensor k is able to compute an upper bound $\hat{p}'_{e,k}$ on $p_{e,k}$ and hence l_k such that $\log_{10} p_{e,k} \leq \log_{10} \hat{p}'_{e,k} \leq \log_{10} \hat{p}_{e,k} = \Delta l_k + \log_{10} p_e^*$. Fortunately, such a $\hat{p}'_{e,k}$ is given by

$$\hat{p}'_{e,k} = Q\left(\frac{a}{2} - \hat{\delta}_k\right) \hat{p}_{e,k-1} + Q\left(\frac{a}{2} + \hat{\delta}_k\right)(1 - \hat{p}_{e,k-1}) \tag{2.26}$$

where

$$\hat{\delta}_k = \frac{1}{a} \ln \left(\frac{1-\hat{p}_{e,k-1}}{\hat{p}_{e,k-1}} \right) \tag{2.27}$$

In fact, since the detection at sensor k is optimal given d_{k-1}, $p_{e,k-1}$, x_k and g_k, the probability error $p_{e,k}$ as given in (2.24) must be non-decreasing function of $p_{e,k-1}$ as long as

$p_{e,k-1} < 0.5$. This fundamental property implies that $\hat{p}'_{e,k}$ given in (2.26) must be no less than $p_{e,k}$ given in (2.24) when $p_{e,k-1} \leq \hat{p}_{e,k-1} < 0.5$.

Performance Illustration

For performance illustration, we use (2.19) and (2.20) and further assume that $g_k = 1$ and $a = 1$. The probability of detection error $p_{e,k}$ versus the sensor index k is illustrated in Figure 2.3.

For the case of "estimation propagation", we used the BLUE estimation where the real-valued estimate $\hat{u}_{k-1}$ and its real-valued variance c_{k-1} are transported from sensor $k-1$ to sensor k. As expected, $p_{e,k}$ converges to zero rapidly.

For the case of "decision propagation - a", the binary decision variable d_{k-1} and the real-valued probability of detection error $p_{e,k-1}$ are transported from sensor $k-1$ to sensor k. It should be noted that $p_{e,k}$ apparently converges to zero when k is very large. (We omitted the portion for $k > 200$.) But a mathematical proof of this property is not yet available.

For the case of "decision propagation - b", four curves are illustrated in this figure. The best (lowest) of the four corresponds to $B = 10$ bits used for quantization of each $p_{e,k-1}$ before the bits are transported from sensor $k-1$ to sensor k. The other three curves correspond to $B = 8$, $B = 6$ and $B = 4$ bits, respectively. We see that the performance of "decision propagation - b" has a nonzero floor when k is beyond a threshold.

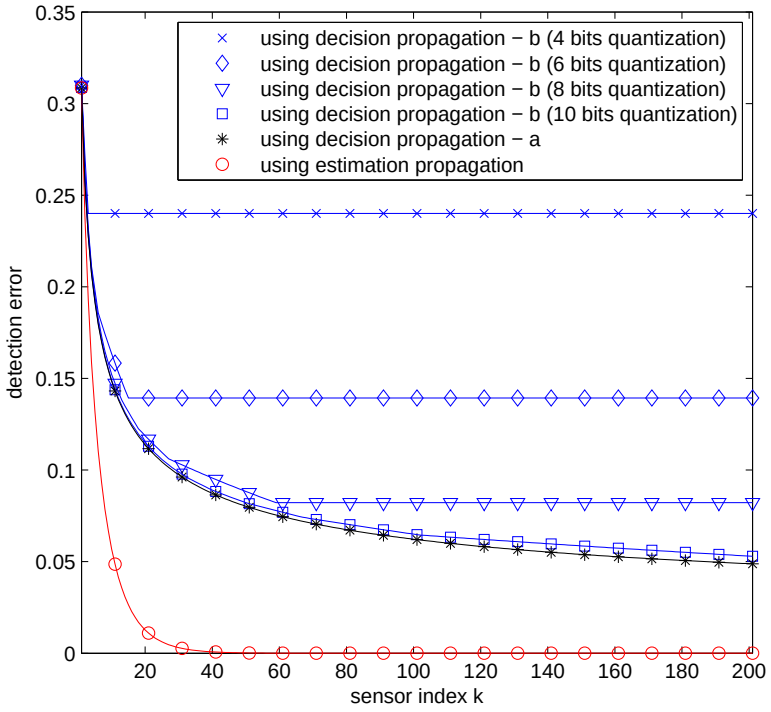

Figure 2.3: The probability of the progressive detection error $p_{e,k}$ versus the sensor index k. The numbers of bits for quantization shown in this plot are used to quantize $p_{e,k-1}$ at sensor $k-1$ nonlinearly. The other parameters used for the quantization are $p_e^* = 0.04$ and $\epsilon = 0.19$.

2.2 Consensus Data Fusion

In previous section, we have described the progressive (multihop) data fusion framework for sensor networks. As we know, there is another type multihop estimation in literature, which is also referred as the consensus data fusion, see [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45] and more references therein. The consensus data fusion is a iterative approach. The concept of consensus data fusion is that each individual sensor refines its local data fusion result by utilizing new

information received from its neighbors. In each round of iteration, each sensor finishes three tasks. Firstly, each sensor, say k-th sensor, listens to its neighbors data transmission and attempts to acquire some innovative information on the desired data. Secondly, each sensor will conduct a local fusion to update its local data fusion result based on its pervious round fusion result and received information innovation from its one hop neighboring sensors. Thirdly, sensor k shares the updated local estimate with its neighbors via wireless communications. The three-step iterative process continues until a consensus of data fusion is reached over the whole network. Here, consensus means that all sensors in the network give the same data fusion result.

Later, we use an example to further illustrate the idea of consensus data fusion. Suppose each sensor i holds an initial scalar value $x_i(0)$. We can think of $x_i(0)$ is the initial estimation of a desired parameter. We apply the consensus data fusion to compute the average of the initial values $\frac{1}{N}\sum_{i=1}^{N} x_i(0)$ where N is the total number of sensors in the network. In each iteration $t+1$, as shown in Fig. 2.4, through the listening phase, sensor k obtained the values of $x_j(t)$ from neighboring sensor $j = j_1, j_2, \cdots, j_5$. Let us denote the neighbors set of sensor k as $\mathcal{N}_k$. Obviously, $N_k = \{j_1, j_2, \cdots, j_5\}$ in this case. Then sensor k conduct a local fusion as below

$$x_k(t+1) = W_{k,k}x_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}x_j(t), \quad k = 1, 2, \cdots, N \tag{2.28}$$

where $W_{k,j}$ is the weight on $\boldsymbol{m}_j$. $W_{k,k}$ is referred as the self-weight. After the local fusion, $x_k t+1$ is transmitted to neighbors of sensor k and it will be utilized in the next round local fusion at those neighbors.

We then write the iterative fusion (2.29) into a compact vector form as

$$\boldsymbol{x}(t+1) = \boldsymbol{W}\boldsymbol{x}(t) \tag{2.29}$$

where $\boldsymbol{x}(t) = (x_1(t), \cdots, x_k(t), \cdots, x_N(t))^T$ and the (i,j)-th entry of $\boldsymbol{W}$ is $W_{i,j}$. It has been shown in [28] that as long as $\boldsymbol{W}$ satisfy three conditions, the above consensus data fusion can lead to $\lim_{t\to\infty} x_k(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$ for all k. In other words, the data fusion results at all sensors converge to a desired common value. The three conditions are listed as follows

$$\mathbf{1}^T\boldsymbol{W} = \mathbf{1}^T \tag{2.30}$$

$$\boldsymbol{W}\mathbf{1} = \mathbf{1} \tag{2.31}$$

$$\rho(\boldsymbol{W} - \mathbf{1}\mathbf{1}^T/N) < 1 \tag{2.32}$$

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

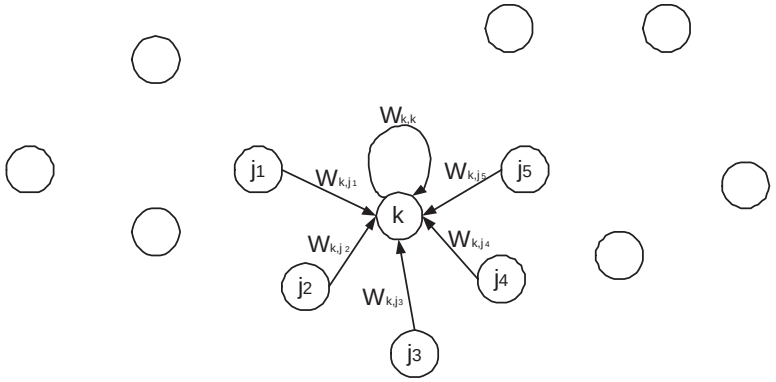

Figure 2.4: The demonstration of the local fusion rule for the consensus based estimation.

2.3 Single-hop Data Fusion

Data fusion based on single hop network structure has been studied extensively by researchers for many years. Here in the section, we brief introduce its principle and review the existing work in this area.

With the single hop data fusion whose system model is shown in Fig. 2.3, the sensor network has a fusion center and each sensor tries to delivery local observations (maybe after local data processing) to the fusion center and the fusion center conducts the final data fusion after gathering sufficient information from sensors. Examples of early work in this category include [6], [7], [9], and references therein. However, most of these works assume 1) the joint distribution of sensor observations is known; 2) the real values of the observations can be transmitted from sensors to a fusion center perfectly without any distortion; 3) The transmission power consumption for sensors is not a big issue worth consideration.

However in practice, the joint distribution of sensor observation noises is very difficult to characterize especially for large scale sensor networks. A class of universal single hop decentralized estimation schemes are then proposed to remove the requirement of the knowledge of joint distribution among sensors [11], [12]. In [11], a universal scheme is proposed to estimate a unknown parameter. The scheme does not need the joint probability distribution of the sensor measurements. The only information it requires is the range of the parameter. By applying this scheme, each sensor transmits a one bit message to a fusion center and the fusion center uses the collected messages to conduct the final estimation. In [12], the universal scheme is extended to the situation where one message could

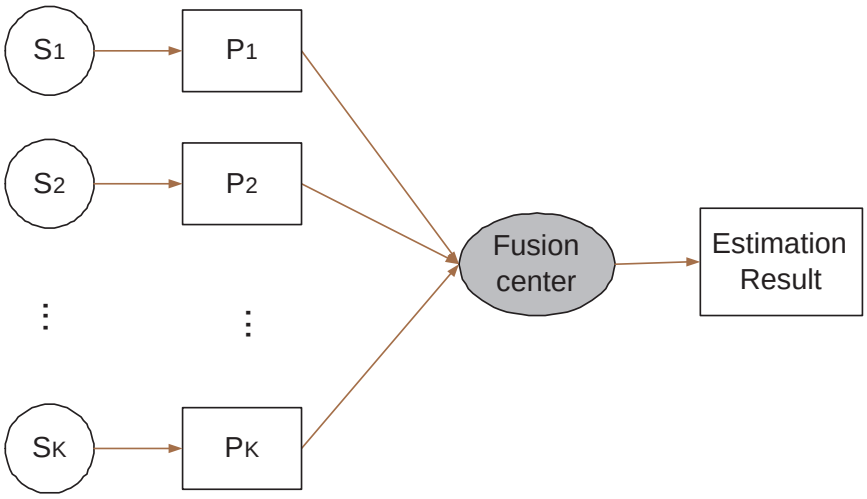

Figure 2.5: The single hop data fusion system model with distributed data observation and processing for a wireless sensor networks. Each circle 'S' stands for one sensor and the box 'P' is the local data processor.

contain multiple bits and the measurement noises is inhomogeneous over sensors. A parallel work on this topic is done in [14] and [15]. A class of ML estimator is studied when only quantized versions (could down to one bit) of the original observations are available under different scenarios from Gaussian noises, to Non-Gaussian noises, and finally to completely unknown noises. In [16], P. Venkitasubramaniam et. al. presented an optimal quantizer design to maximize the minimum asymptotic relative efficiency between the quantized and unquantized maximum likelihood (ML) estimators in wireless sensor networks.

The above schemes take care of the situation of without joint probability distribution information of sensor observations, and also consider the quantization of the original values of observations. However, These works still assume that the wireless links between sensors and the fusion center are perfect. Hence transmissions of observation are also assumed perfect from sensors to the fusion center. But due to the power limit of sensors

and the fading of wireless channels, these assumptions of idea channel and perfect data transmission are unrealistic in practice. To combat with wireless channel fading and noise, we need to incorporate practical data communication methods and consider transmission energy planning with them.

In recent papers of [13], [17], [18], and [19], the single hop distributed estimation with realistic data transmission and energy planning are considered. There are two data transmission approaches we could consider, namely, the analog transmission and the digital transmission. The analog transmission amplifies the local observations at each sensor then transmit them via analog modulation to the fusion center, which we refer to as the amplify-and-forward approach. The purpose of the signal amplifying is to combat against the additive noise and wireless channel fading as well. [19] has studied the method of using this amplify-and-forward approach to estimate an unknown scalar parameter, based on the transmission energy minimization criteria. [17] and [18] then extend the amplify-and-forward approach into the context of vector estimation. On the other hand, the digital transmission is to quantize the observations into bits (possibly down to only one bit) and then transmits those bits to the fusion center using digital communication technologies [13]. We name this approach as quantize-then-transmit, in contrast to the amplify-and-forward approach. The optimal quantization bits allocation is proposed in [13] to minimize the total transmission energy, while guarantee the estimation MSE at the fusion center at the same time.

2.4 Summary

In this chapter, we have introduced the concepts of progressive data fusion, consensus data fusion, and single-hop data fusion. Progressive data fusion framework is the original contribution of this dissertation. The classical estimation principles: BLUE and LMMSE, have been shown to be naturally embeddable into progressive data fusion. The progressive detection based on estimation propagation appears promising in both theory and practice.

Chapter 3

Energy Planning for Progressive Data Fusion

3.1 System Model

In this section, we consider a wireless sensor network where each sensor can sense, compute, transmit and receive. The data collected by each sensor $\boldsymbol{x}_k^{(n)}$ can be of any dimension and any statistical properties. But we assume that each sensor (sensor k) is able to use its collected data to obtain an estimate $\boldsymbol{\theta}_k^{(n)} \in R^{M\times 1}$ of a desired unknown deterministic vector denoted by $\boldsymbol{\theta}^{(n)} \in R^{M\times 1}$, where n denotes the sampling time. We assume that $\boldsymbol{\theta}_k^{(n)}$ is unbiased and has a bounded covariance matrix, i.e., $E\{\boldsymbol{\theta}_k^{(n)}\} = \boldsymbol{\theta}^{(n)}$ and $E\{\boldsymbol{\theta}_k^{(n)}\boldsymbol{\theta}_k^{(n)^T}\} \leq \boldsymbol{C}_{\theta,k}$. And furthermore, $E\{\boldsymbol{\theta}_k^{(n)}\boldsymbol{\theta}_i^{(l)^T}\} = 0 \ \forall \ k \neq i$. We assume that the estimation of $\boldsymbol{\theta}^{(n)}$ is done independently of that of $\boldsymbol{\theta}^{(l)}$ for $l \neq n$. And all data observation, computations, and communications required for estimating $\boldsymbol{\theta}^{(n)}$ are performed within the

time window from the sampling time n to the sampling time $n+1$. Because of this reason, we will drop the index n in $\boldsymbol{x}_k^{(n)}$ and $\boldsymbol{\theta}^{(n)}$ for convenience if no confusion will be caused.

The estimation of the unknown parameter vector $\boldsymbol{\theta}$ is done in the progressive fashion as described in chapter 2. We assume that there is a routing tree from all sensors towards a destination node as illustrated in Figure 3.1. There are many ways to establish a routing tree. One example is the technique for finding Steiner tree or the minimum distance tree [46]. But the minimum distance tree may not be optimal for the purpose of distributed estimation. Finding the best tree for distributed data fusion still remains a open problem for research in this area. The destination node in the routing tree may also act as a scheduler which knows the observation matrices of all sensors, plans the transmission energy to be consumed by each sensor, and broadcasts all essential information to all sensors. The purpose of the transmission energy planning is to ensure a desired estimation performance at the destination node while keeping the total energy cost as low as possible. The process of routing tree establishment, transmission energy planning and decision broadcast are all done at a startup of the network or when the network topology changes. With the decision from the scheduler, each sensor in the network performs an estimation of $\boldsymbol{\theta}$ using both its own observation $\boldsymbol{x}_k$ and the estimates obtained by its upper stream sensors.

We assume that the time interval between $\boldsymbol{x}_k^{(n)}$ and $\boldsymbol{x}_k^{(n+1)}$ is sufficiently long so that during this time interval all communications between sensors and all computations at each sensor can be completed for the estimation of $\boldsymbol{\theta}^{(n)}$ at the destination node. We assume that orthogonal or approximately orthogonal scheduling is made for all links in the network. Since only neighboring nodes are communicating with each other, there can be

concurrent and co-channel transmissions without major interferences to each other. We also assume that the network (including network topology and all channel state information) is sufficiently stationary so that each energy planning cycle can span a time window of at least many time instants of $\boldsymbol{\theta}^{(n)}$ and hence the overhead of the energy planning is acceptable.

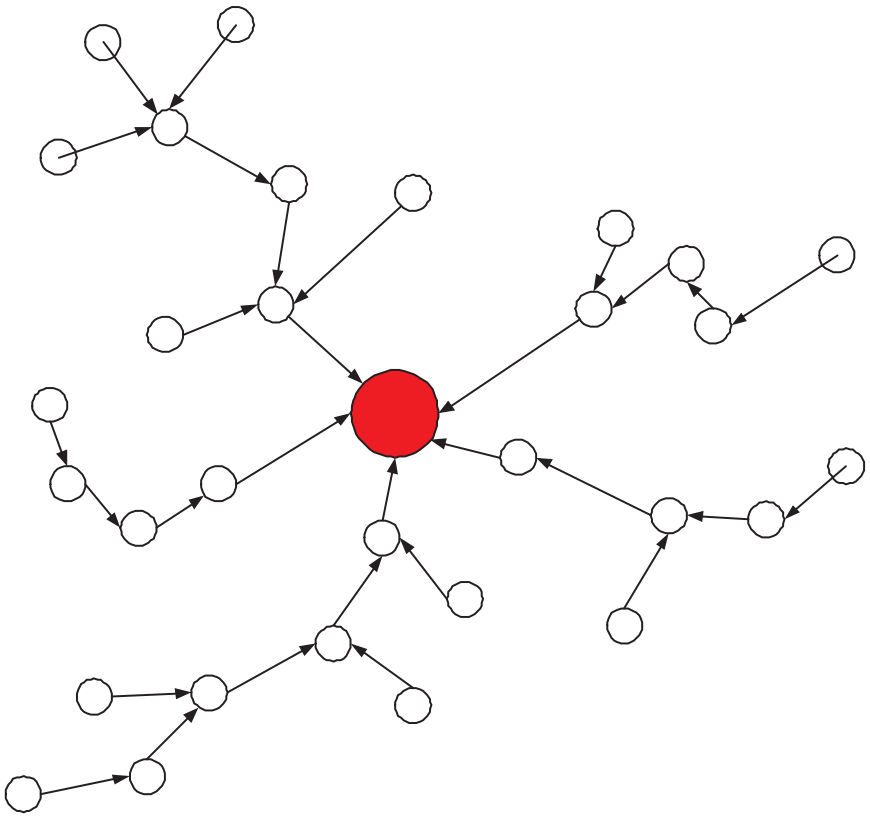

Figure 3.1: The snapshot of a network tree applying the progressive estimation. The solid biggest node in the middle is the destination node.

3.2 Communication Channel Model

Both analog communication and digital communication have been considered in the literature on sensor network signal processing. In this section, we will provide a com-

parison of energy consumption between the two. In order to do so, we need to first establish the energy model for each communication mode as follows.

We denote by $\mathcal{T}$ and $\mathcal{W}$ respectively the total time and bandwidth available for communication for each link and each observation sample (i.e., for each k and n in $x_k(n)$). We assume that the radio frequency (RF) channels between neighboring sensors have constant channel gains within the bandwidth $\mathcal{W}$ and a whole cycle of transmission energy planning. For a RF channel of single-sided bandwidth $\mathcal{W}$, there is an equivalent complex baseband channel of the double-sided bandwidth $\mathcal{W}$, which corresponds to a pair of in-phase and quadrature channels.

We let h be the complex baseband channel gain from a sensor to its downstream sensor. The channel noise is assumed to be Gaussian with the power spectral density N_0 within the baseband $[-\mathcal{W}/2, \mathcal{W}/2]$.

To transmit a complex symbol s from one sensor to another in the analog mode, we need to use s to modulate (via amplitude, phase, frequency, pulse width, pulse position, etc.) a waveform $c(t)$ which has a time duration effectively no larger than $\mathcal{T}$ and a bandwidth effectively no larger than $\mathcal{W}$. Here, we use the word "effectively" because if zero error is tolerated, then the time duration of any waveform must be infinite if the bandwidth is finite and vice versa. A fundamental constraint on $\mathcal{T}$ and $\mathcal{W}$ is $\mathcal{TW} \geq 1$. At the output of the baseband channel, a demodulation is performed to obtain an estimate $\hat{s}$ of s. The signal to noise ratio (SNR) of the estimated symbol at the channel output is

$$SNR = \frac{|s|^2}{\mathcal{E}\{|\hat{s} - s|^2\}} = \mu |h|^2 E/N_0 \tag{3.1}$$

where $\mathcal{E}$ denotes expectation, E is the total transmitted energy and μ ($\mu > 0$) is a parameter

depending on the modulation method. As shown in Appendix 3.11.1, if the pulse amplitude modulation and the matched filtering demodulation are used, then $\mu = 1$ and $E = |s|^2 E_c$ where E_c is the energy of the waveform $c(t)$.

Note that the combination of the waveform modulator, the continuous-time baseband channel and the waveform demodulation (with sampling) constitute a discrete-time analog channel, which is our model for analog communication of discrete-time symbols. For background knowledge of modulation, demodulation and sampling, there are many textbooks such as [47].

In order to transmit a complex symbol from one sensor to another in the digital mode, we can use the above described discrete-time analog channel preceded by a digital encoder and followed by a digital decoder [47]. Furthermore, if the time-bandwidth product of the RF channel is such that $\mathcal{TW} \geq L \geq 1$ where L is an integer, then we can partition the original channel into L sub-channels (in frequency and/or time). If a MIMO channel is used, the number of sub-channels is increased by a factor equal to the rank of the channel matrix for the same time-bandwidth product. For each sub-channel, we can apply the above mentioned modulation method to yield an discrete-time analog channel with an output SNR which is proportional to the input energy of the sub-channel. In other words, with $\mathcal{TW} \geq L \geq 1$, we can construct L discrete-time analog sub-channels and the lth channel has the output SNR: $SNR_l = \mu |h|^2 E/(LN_0)$ where E denotes the total transmission energy over the L sub-channels. Then, the total number of bits that we can transmit digitally over

the RF channel within the time $\mathcal{T}$ and bandwidth $\mathcal{W}$ (with negligible errors) is

$$B = \sum_{l=1}^{L} \psi \log_2(1 + \phi SNR_l) = \psi L \log_2 \left(1 + \phi\mu \frac{|h|^2 E}{LN_0}\right) \tag{3.2}$$

where ψ $(0 < \psi < 1)$ and ϕ $(0 < \phi < 1)$ are penalty factors depending on the coding method [48] and packet overhead. This model is appropriate even if the coding is done within a single observation time interval of each link. Alternatively, we can write the transmission energy in terms of the number of bits as follows:

$$E = \frac{LN_0}{\phi\mu|h|^2} \left(2^{\frac{B}{\psi L}} - 1\right) \tag{3.3}$$

The model (3.3) will be used to design the transmission energy planning for digital communication between sensors. The model (3.1) will be used for the analog case. The energy planning problem will be formulated as minimization of the transmission energy within the entire network subject to that the mean square error (MSE) of the estimate at the destination node is no larger than a pre-specified value.

3.3 Probabilistic Uniform Quantization

The communication between adjacent sensors can be digital or analog in principle. When the digital communication is applied, we assume that each sensor quantizes its local estimate using a uniform quantization method. For example, if $\hat{\theta}$ is a local estimate of a scalar parameter θ and is bounded between $[W, -W]$, the quantization of $\hat{\theta}$ is uniform and probabilistic as follows. Let B be the number of bits used to quantize $\hat{\theta}$ within $[W, -W]$, and $\Delta = \frac{2W}{2^B - 1}$ be the quantization interval. If $-W + i\Delta \leq \hat{\theta} \leq -W + (i+1)\Delta$ and $\hat{\theta} + W - i\Delta = r\Delta$, then $\hat{\theta}$ is quantized to $-W + i\Delta$ with the probability $1 - r$ and to

$-W + (i+1)\Delta$ with the probability r. As shown in [12], it is easy to verify that the error of this quantization method has zero mean and the variance σ_q^2 bounded as follows:

$$\sigma_q^2 \leq \max_r r(1-r)\Delta^2 = \frac{\Delta^2}{4} \leq \frac{4W^2}{2^{2B}} \tag{3.4}$$

where the last inequality holds under $B \geq 1$. Note that if $\hat{\theta}$ is uniformly distributed within $[W, -W]$, then $\sigma_q^2 = \int_0^1 r(1-r)\Delta^2 dr = \frac{\Delta^2}{6}$.

We will not consider more advanced quantization methods such as vector quantization, or quantization based on a statistical knowledge of the random variables. The variance expression of a practical vector quantizer is generally difficult to get. Even for a single Gaussian random variable, the variance expression of the optimal quantization errors is not expressive enough for our purpose, e.g., see [49]. The simple distortion-rate expression in [50] requires an ideal vector quantizer of many observation samples, which is hard to implement. A recent work on source quantization in the context of sensor networks with structured trees is available in [51] where the knowledge of the joint statistical distribution of the source and the side data is required.

3.4 Analog Transmission versus Digital Transmission

Digital communication has many advantages over analog communication, which includes modularity and robustness. These two advantages appear very important for large-scale multihop wireless networks. But from the pure transmission energy point of view, the conclusion is not always in favor of one over the other, which depends on the available time-bandwidth product.

Transmitting a complex symbol s from one sensor to another by the analog mode introduces an additional noise term whose variance is inversely proportional to the transmission energy E according to (3.1). More specifically, the variance of the additional noise caused by the analog communication is

$$\sigma^2_{analog} = \frac{|s|^2}{SNR} = \frac{|s|^2 N_0}{\mu |h|^2 E} \tag{3.5}$$

We now consider transmitting the same complex symbol s from one sensor to another by the digital mode using the same amount of total energy E. For digital transmission, there is an additional noise due to quantization. The variance of this quantization error using total B bits for both real part and imaginary part (each bounded between $[-W, W]$) is upper bounded as follows according to (3.4) and (3.2):

$$\sigma^2_{digital} \leq \frac{8W^2}{2^B} = \frac{8W^2}{\left(1 + \phi\mu \frac{|h|^2 E}{LN_0}\right)^{\psi L}} \tag{3.6}$$

We see that if $\psi L \leq 1$, the exact value of $\sigma^2_{digital}$ (which depends on the quantization method) can be larger than σ^2_{analog} even when the transmission energy E is large. But if $\psi L > 1$, the upper bound of $\sigma^2_{digital}$ decreases faster than σ^2_{analog} as E increases. Furthermore, since $\lim_{m\to\infty}(1 + 1/m)^m = e$, if L is large, we can write

$$\sigma^2_{digital} \leq 8W^2 \exp\left\{-\psi\phi\mu \frac{|h|^2 E}{N_0}\right\} \tag{3.7}$$

Here, we see that the variance of the quantization error decreases exponentially as the transmission energy increases.

We conclude that as long as there is a sufficient transmission time-bandwidth product for each link and each observation sample, the digital transmission is more efficient

in transmission energy than the analog transmission. The above analysis also suggests that otherwise the analog transmission can be more efficient than the digital transmission. It should be useful to note that an advantage of analog transmission over digital transmission, as advocated in [52] and [53], was based on a single-hop network, a fixed transmission time-bandwidth and an increasing number of sensors. Their results and ours are complementary to each other. The analog-digital comparison shown in [54] assumes the use of a single channel for digital transmission, and the authors overlooked the importance of sub-division of time-bandwidth product for digital transmission.

3.5 Progressive Estimation with Digital Transmission

We will let K be the total number of sensors in the network, I_k be the number of the upper stream sensors of sensor k, and S_k be the set of the indices of the nodes that are the upper stream sensors of sensor k. Each sensor is assumed to be connected to a destination node via a routing tree, see Figure 3.1.

Since the estimation of the unknown vector $\boldsymbol{\theta}^{(n)}$ is done for each sampling time instant n and its procedure is the same for all n, we will remove n for convenience. At each sensor k, the estimation of the unknown vector $\boldsymbol{\theta}$ is based on the local observation $\boldsymbol{x}_k$ and the quantized estimates $\boldsymbol{m}_i$, $i \in S_k$, from the upper stream sensors of sensor k.

3.5.1 BLUE

Suppose that $\boldsymbol{\theta}$ is an unknown deterministic vector, and $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\theta} + \boldsymbol{n}$ is the available data vector, where $\boldsymbol{M}$ is a matrix of full column rank and $\boldsymbol{n}$ is a noise vec-

tor of zero mean and the covariance matrix $\boldsymbol{C}$. Then, the best linear unbiased estimate (BLUE) or equivalently the minimum variance unbiased linear estimate of $\boldsymbol{\theta}$ is known as $\hat{\boldsymbol{\theta}} = \left(\boldsymbol{M}\boldsymbol{C}^{-1}\boldsymbol{M}\right)^{-1}\boldsymbol{M}\boldsymbol{C}^{-1}\boldsymbol{y}$, and the covariance matrix of this estimate is $\left(\boldsymbol{M}\boldsymbol{C}^{-1}\boldsymbol{M}\right)^{-1}$, e.g., see [55].

We now let $\boldsymbol{m}_k$ be an unbiased quantized BLUE estimate at sensor k and $\hat{\boldsymbol{C}}_k$ be the covariance matrix of $\boldsymbol{m}_k$. If $\boldsymbol{m}_i$ and $\hat{\boldsymbol{C}}_i$ for $i \in S_k$ are available at sensor k, the BLUE of $\boldsymbol{\theta}$ at sensor k is given by

$$\hat{\boldsymbol{\theta}}_k = \left(\boldsymbol{G}_k^H\boldsymbol{G}_k + \sum_{i\in S_k}\hat{\boldsymbol{C}}_i^{-1}\right)^{-1}\left(\boldsymbol{G}_k^H\boldsymbol{x}_k + \sum_{i\in S_k}\hat{\boldsymbol{C}}_i^{-1}\boldsymbol{m}_i\right) \tag{3.8}$$

and its covariance matrix is $\mathcal{E}\left\{(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta})^H\right\} = \left(\boldsymbol{G}_k^H\boldsymbol{G}_k + \sum_{i\in S_k}\hat{\boldsymbol{C}}_i^{-1}\right)^{-1}$. Here, $\mathcal{E}$ denotes expectation and the superscript ^{H} denotes complex conjugate transpose.

Assume that both the real and image part of the mth element of $\hat{\boldsymbol{\theta}}_k$ is bounded between $[-W_m, W_m]$ and quantized using total $B_{k,m}$ bits. (Note that it is meaningful to have a different range for a different element of $\hat{\boldsymbol{\theta}}_k$. This range should be governed by the prior knowledge of the physical nature and resolution requirement of each element of $\boldsymbol{\theta}_k$, which could be temperature, wavelength, distance, etc.) Then the covariance matrix $\hat{\boldsymbol{C}}_k$ of the quantized BLUE estimate $\boldsymbol{m}_k$ at sensor k is bounded as follows:

$$\hat{\boldsymbol{C}}_k \leq \left(\boldsymbol{G}_k^H\boldsymbol{G}_k + \sum_{i\in S_k}\hat{\boldsymbol{C}}_i^{-1}\right)^{-1} + \boldsymbol{C}_{q,k} \tag{3.9}$$

where the mth diagonal element of $\boldsymbol{C}_{q,k}$ is denoted by $\bar{\sigma}^2_{q,k,m}$ and given by $\bar{\sigma}^2_{q,k,m} = \frac{8W_m^2}{2^{B_{k,m}}}$. However, except for the upper bound, the exact covariance matrix of the quantized estimate at each sensor is difficult to keep track of. Hence, the exact BLUE is not feasible for our

application. Nevertheless, the above discussion is an important part of our systematic treatment.

3.5.2 Quasi BLUE

Since the exact $\hat{\boldsymbol{C}}_i$ for $i \in S_k$ are not available at sensor k, we now replace them by their upper bounds $\bar{\boldsymbol{C}}_i$ for $i \in S_k$. Using these upper bounds, the following estimate will be referred to as quasi BLUE at sensor k:

$$\bar{\boldsymbol{\theta}}_k = \left(\boldsymbol{G}_k^H \boldsymbol{G}_k + \sum_{i \in S_k} \bar{\boldsymbol{C}}_i^{-1} \right)^{-1} \left(\boldsymbol{G}_k^H \boldsymbol{x}_k + \sum_{i \in S_k} \bar{\boldsymbol{C}}_i^{-1} \boldsymbol{m}_i \right) \tag{3.10}$$

and the covariance matrix of this estimate is upper bounded by $\left(\boldsymbol{G}_k^H \boldsymbol{G}_k + \sum_{i \in S_k} \bar{\boldsymbol{C}}_i^{-1} \right)^{-1}$. Consequently, we can compute the upper bound $\bar{\boldsymbol{C}}_k$ of the covariance matrix of the quantized quasi-BLUE estimate $\boldsymbol{m}_k$ by

$$\bar{\boldsymbol{C}}_k = \left(\boldsymbol{G}_k^H \boldsymbol{G}_k + \sum_{i \in S_k} \bar{\boldsymbol{C}}_i^{-1} \right)^{-1} + \boldsymbol{C}_{q,k} \tag{3.11}$$

Our goal is to determine $B_{k,m}$ for all k and m such that the total transmission energy can be significantly reduced subject to a MSE constraint at the destination node. But according to the recursion of the covariance matrices (3.11) of the quasi-BLUE, the MSE at the destination is a very complicated function of $B_{k,m}$. Unless the network is very small, finding the optimal $B_{k,m}$ for quasi-BLUE is not feasible.

3.5.3 Averaging

There is a simpler method for progressive estimation, i.e., taking a simple average of the quantized estimates from upper stream sensors together with the BLUE estimate

solely based on $\boldsymbol{x}_k$. This estimate at sensor k is given by:

$$\tilde{\theta}_k = \frac{1}{I_k+1}(\boldsymbol{G}_k^H \boldsymbol{G}_k)^{-1}\boldsymbol{G}_k^H \boldsymbol{x}_k + \frac{1}{I_k+1}\sum_{i\in S_k}\boldsymbol{m}_i \tag{3.12}$$

If the mth element of $\tilde{\theta}_k$ is quantized with $B_{k,m}$ bits and the upper bounds of the covariance matrices of $\boldsymbol{m}_i$ are denoted by $\widetilde{\boldsymbol{C}}_i$, then the corresponding upper bound $\widetilde{\boldsymbol{C}}_k$ of the covariance matrix of the quantized estimate $\boldsymbol{m}_k$ at sensor k is given by

$$\widetilde{\boldsymbol{C}}_k = \frac{1}{(I_k+1)^2}(\boldsymbol{G}_k^H \boldsymbol{G}_k)^{-1} + \frac{1}{(I_k+1)^2}\sum_{i\in S_k}\widetilde{\boldsymbol{C}}_i + \boldsymbol{C}_{q,k} \tag{3.13}$$

Note that the estimate given by (3.12) does not need the covariance matrices of the estimates from upper stream sensors. Also note that the recursion of the covariance matrices given by (3.13) is much simpler than that in (3.11). It follows from the following lemma that $\widetilde{\boldsymbol{C}}_k$ is simply an upper bound of $\bar{\boldsymbol{C}}_k$, i.e., $\widetilde{\boldsymbol{C}}_k \geq \bar{\boldsymbol{C}}_k$. But as implied by a simulation example shown later, this upper bound is quite loose generally.

Lemma 1 *Given n positive definite matrices $\boldsymbol{A}_k, k = 1, \cdots, n$, we have*

$$\left(\sum_{k=1}^{n}\boldsymbol{A}_k^{-1}\right)^{-1} \leq \frac{1}{n^2}\left(\sum_{k=1}^{n}\boldsymbol{A}_k\right) \tag{3.14}$$

Proof. See Appendix 3.11.2. ■

3.6 Energy Planning with Digital Transmission

3.6.1 Algorithm 1

In this section, we design an algorithm for computing the bit allocations among all sensors. To design such an algorithm, formulating a tractable optimization problem is essential.

The total number of bits used for the quantized estimate at sensor k is $\sum_{m=1}^{M} B_{k,m}$, where $B_{k,m}$ is the number of bits for the mth element of the estimated vector at sensor k. In algorithm 1, we do not consider adaptive allocation of quantization bits over sub-channels. Therefore each sub-channel will need to transmit $\sum_{m=1}^{M} B_{k,m}/L$ number of bits. From the energy model (3.3), the transmission energy to be consumed by sensor k is given by

$$E_k = a_k \left(2^{\frac{1}{\psi L} \sum_{m=1}^{M} B_{k,m}} - 1 \right) \tag{3.15}$$

where $L = \lfloor \mathcal{TW} \rfloor \geq 1$, $a_k = \sum_{l=1}^{L} \frac{N_0}{\phi \mu |h_{k,l}|^2}$, and $|h_{k,l}|^2$ is the squared channel gain for sub-channel l from sensor k to its down stream sensor.

For any n positive real numbers b_i, it is known that $\prod_{i=1}^{n} b_i \leq \sum_{i=1}^{n} b_i^n/n$. It follows that

$$E_k < a_k 2^{\frac{1}{\psi L} \sum_{m=1}^{M} B_{k,m}} \leq \frac{a_k}{M} \sum_{m=1}^{M} 2^{\frac{M}{\psi L} B_{k,m}} = \sum_{m=1}^{M} \bar{E}_{k,m} \tag{3.16}$$

where $\bar{E}_{k,m} = \frac{a_k}{M} 2^{\frac{M}{\psi L} B_{k,m}}$, which can be viewed as the upper bound of the energy needed to transmit the mth element of the quantized estimation. We now define the following cost function:

$$J_p = \sum_{k=1}^{K-1} \sum_{m=1}^{M} \bar{E}_{k,m}^{p} = \sum_{k=1}^{K-1} \sum_{m=1}^{M} \left(\frac{a_k}{M} \right)^p 2^{\frac{pM}{\psi L} B_{k,m}} \tag{3.17}$$

which is the pth power of the L_p norm of $\bar{E}_{k,m}$ for all k and all m. Note that at the destination node which is labelled as node K, there is no need for quantization. Clearly, when $p = 1$, we have $J_1 = \sum_{k=1}^{K-1} \sum_{m=1}^{M} \bar{E}_{k,m}$, which is an upper bound on the total (sum) transmission energy.

To formulate the MSE constraint at the destination node, we use the linear recur-

sion (3.13) of the covariance matrices. It follows from (3.13) that

$$\widetilde{\boldsymbol{C}}_K = \sum_{k=1}^{K} \frac{\alpha_k}{(I_k+1)^2} (\boldsymbol{G}_k^H \boldsymbol{G}_k)^{-1} + \sum_{k=1}^{K-1} \alpha_k \boldsymbol{C}_{q,k} \tag{3.18}$$

where $\alpha_K = 1$, and $\alpha_i = \frac{\alpha_k}{(I_k+1)^2}$ for $i \in S_k$. Note that in order to compute α_k for all k, one should start with the sensors nearest to the destination node and then proceed outwards recursively. The actual values of α_k depend on the routing tree. Then, the MSE at the destination node is given by

$$MSE_K = tr(\widetilde{\boldsymbol{C}}_K) = \xi + \sum_{k=1}^{K-1} \sum_{m=1}^{M} \alpha_k \frac{4W_m^2}{2^{B_{k,m}}} \tag{3.19}$$

where $\xi = \sum_{k=1}^{K} \frac{\alpha_k}{(I_k+1)^2} tr\left((\boldsymbol{G}_k^H \boldsymbol{G}_k)^{-1}\right)$ which is invariant to $B_{k,m}$. If MSE_0 is the desired MSE value at the destination node, it is meaningful to set up the following constraint

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \alpha_k \frac{W_m^2}{2^{B_{k,m}}} \leq \eta \tag{3.20}$$

where $\eta = \frac{1}{4}(MSE_0 - \xi)$.

The problem now can be formulated as below

$$\min_{B_{k,m}} : J_p = \sum_{k=1}^{K-1} \sum_{m=1}^{M} \left(\frac{a_k}{M}\right)^p 2^{\frac{pM}{\psi L} B_{k,m}} \tag{3.21}$$

subject to:

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \alpha_k \frac{W_m^2}{2^{B_{k,m}}} \leq \eta \tag{3.22}$$

$$B_{k,m} \geq 0 \tag{3.23}$$

To solve this problem, we need the Hölder's inequality: For $x_i > 0$, $y_i > 0$ where $i = 1, 2, \cdots, n$, if $k > 1$, and $\frac{1}{k} + \frac{1}{k'} = 1$, then,

$$\left(\sum_{i=1}^{n} x_i^k\right)^{\frac{1}{k}} \left(\sum_{i=1}^{n} y_i^{k'}\right)^{\frac{1}{k'}} \geq \sum_{i=1}^{n} x_i y_i \tag{3.24}$$

where the equality holds if $x_i^{k-1} = \lambda y_i$ for some constant λ and all i. The equality condition is easy to verify by recognizing $(k-1)k' = k$.

By defining $A_{k,m} = \left((a_k/M)^p 2^{\frac{Mp}{\psi L} B_{k,m}}\right)^{\frac{\psi L}{\psi L+Mp}}$ and $C_{k,m} = \left(\frac{\alpha_k W_m^2}{2^{B_{k,m}}}\right)^{\frac{Mp}{\psi L+Mp}}$, the Hölder's inequality implies

$$\begin{aligned}\left\{\sum_{k=1}^{K-1}\sum_{m=1}^{M} A_{k,m}^{\frac{\psi L+Mp}{\psi L}}\right\}^{\frac{\psi L}{\psi L+Mp}} \left\{\sum_{k=1}^{K-1}\sum_{m=1}^{M} C_{k,m}^{\frac{\psi L+Mp}{Mp}}\right\}^{\frac{Mp}{\psi L+Mp}} &\geq \sum_{k=1}^{K-1}\sum_{m=1}^{M} A_{k,m}C_{k,m} \\ &= \sum_{k=1}^{K-1}\sum_{m=1}^{M} (a_k/M)^{\frac{\psi Lp}{\psi L+Mp}} (\alpha_k W_m^2)^{\frac{Mp}{\psi L+Mp}} \\ &\triangleq \gamma \end{aligned} \tag{3.25}$$

where the equality holds when

$$A_{k,m}^{\frac{Mp}{\psi L}} = \lambda C_{k,m}. \tag{3.26}$$

It is important to note that γ is independent of $B_{k,m}$. The constraint (3.36) is equivalent to $\sum_{k=1}^{K-1}\sum_{m=1}^{M} C_{k,m}^{\frac{\psi L+Mp}{Mp}} \leq \eta$. Then, using (3.17), (3.36) and (3.25), we have

$$J_p = \left\{\sum_{k=1}^{K-1}\sum_{m=1}^{M} A_{k,m}^{\frac{\psi L+Mp}{\psi L}}\right\} \geq \frac{\gamma^{\frac{\psi L+Mp}{\psi L}}}{\eta^{\frac{Mp}{\psi L}}} \tag{3.27}$$

The right hand side of (3.27) is independent of $B_{k,m}$ and is also the minimum of J_p. This minimum is achieved if (3.26) holds and the equality in (3.36) holds. But (3.26) along with the constraint $B_{k,m} \geq 0$ implies that

$$B_{k,m} = \frac{\psi L}{Mp\psi L + M^2p^2}\left((\psi L + Mp)\log_2\lambda + 2Mp\log_2\left(\sqrt{\alpha_k}W_m\right) - Mp^2\log_2\frac{a_k}{M}\right)^+ \tag{3.28}$$

where $(x)^+ = \max(0, x)$. Using (3.28) in the equality of (3.36) yields

$$\lambda = \frac{1}{\eta^{\frac{Mp}{\psi L}}}\left(\sum_{(k,m)\in S^+} (\sqrt{\alpha_k}W_m)^{\frac{2Mp}{\psi L+Mp}} \left(\frac{a_k}{M}\right)^{\frac{p\psi L}{\psi L+Mp}}\right)^{\frac{Mp}{\psi L}} \tag{3.29}$$

where $S^+ = \{(k,m)|B_{k,m} > 0\}$. The two equations (3.28) and (3.29) need to be computed iteratively until convergence. The iteration starts with the computation of λ from (3.29) with a full set S^+ where $B_{k,m} > 0$ for all k and m. After convergence, each $B_{k,m}$ is rounded up into an integer.

We now show that the number of iterations between (3.28) and (3.29) until convergence is finite. For any given λ, the solution of $B_{k,m}$ from (3.28) is unique. If none of $B_{k,m}$ is zero after the first iteration, the convergence is achieved. If one or more of $B_{k,m}$ become zero, the set S^+ is reduced which in turn reduces λ via (3.29). From (3.28), we see that $B_{k,m}$ does not increase as λ decreases, which means that the previous zero $B_{k,m}$ will remain zero after another iteration. If there is no additional $B_{k,m}$ becoming zero at the end of an iteration, the size of S^+ is not changed and hence the convergence is achieved. Since the size of the initial S^+ is finite, so is the number of iterations required for convergence. It is obvious that the converged solution is invariant to the initial choices of $B_{k,m}$ as long as they are positive for all k and s.

It is also useful to note that (3.28) and (3.29) constitute a "water-filling" type algorithm. In other words, (3.28) can be compressed into the form $B_{k,m} = (l - s_{k,m})^+$ where l resembles a "water level" that is independent of the location parameters k and m in a "water tank", $s_{k,m}$ resembles the heights of the steps at the bottom of the water tank, and $B_{k,m}$ is the depth of the water at the location k and m.

The number L of the sub-channels is of great importance to the energy planning. From (3.3) or (3.15), one can readily verify that the transmission energy E is a decreasing function of L. However, the effect of increasing L on energy saving diminishes at large L.

Taking the limit $L \to \infty$, the energy model (3.15) becomes

$$E_k = \frac{N_0}{\phi\psi\mu|h_k|^2} \ln 2 \sum_{m=1}^{M} B_{k,m} \tag{3.30}$$

which is invariant to L. Furthermore, as $L \to \infty$, (3.28) and (3.29) can be rewritten as follows:

$$B_{k,m} = \left(\log_2 \lambda' + 2 \log_2 \left(\sqrt{\alpha_k} W_m \right) - p \log_2 \frac{a'_k}{M} \right)^+ \tag{3.31}$$

$$\lambda' = \frac{1}{\eta} \sum_{(k,m) \in S^+} \left(\frac{a'_k}{M} \right)^p \tag{3.32}$$

where $a'_k = \frac{N_0}{\phi\psi\mu|h_k|^2}$. Both (3.31) and (3.32) are independent of L. Therefore, the energy planning is virtually invariant to L when L is large. The simulation results shown later suggest that $L \geq 4M$ is large enough for the energy planning to become practically invariant to L.

Another important factor is the norm p used to formulate the cost function (3.17). Selecting $p = 1$ is to minimize the total energy of the network. Using a larger p implies that we want to penalize the larger energy terms $\bar{E}_{k,m}$. In the extreme case where $p = +\infty$, we minimize the maximum value among $\bar{E}_{k,m}$. By taking $p \to \infty$, (3.28) and (3.29) can be rewritten as follows:

$$B_{k,m} = \left(\log_2 \lambda'' - \frac{\psi L}{M} \log_2 \frac{a_k}{M} \right)^+ \tag{3.33}$$

$$\lambda'' = \frac{1}{\eta} \sum_{(k,m) \in S^+} \alpha_k W_m^2 \left(\frac{a_k}{M} \right)^{\frac{\psi L}{M}} \tag{3.34}$$

It then follows that unless $B_{k,m} = 0$, $\bar{E}_{k,m} = \frac{a_k}{M} 2^{\frac{M}{\psi L} B_{k,m}} = (\lambda'')^{\frac{M}{\psi L}}$ which is independent of k and m.

3.6.2 Algorithm 2

In algorithm 2, again we assume that there are L communication sub-channels from any sensor to its down-stream sensor within each time window. Unlike algorithm 1, which utilizes even bits allocation over communication sub-channels, we consider optimal bits distribution over sub-channels in addition to optimal quantization bits allocation at each sensor k. Specifically, we let $B'_{k,l}$ be the number of bits transmitted from sensor k to its down-stream sensor in sub-channel l. It follows that $\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l}$. Note that while the final solution for $B_{k,m}$ must be integer, the individual $B'_{k,l}$ does not have to be integer because the coding for the sub-channels can be done jointly.

Similarly as in algorithm 1, the energy cost for transmitting $B'_{k,l}$ from sensor k to its down-stream sensor in sub-channel l can be modelled as:

$$E_{k,l} = a_{k,l}\left(2^{B'_{k,l}/\varphi} - 1\right) \tag{3.35}$$

where $a_{k,l} = \frac{N_0}{\phi\mu|h_{k,l}|^2}$, N_0 is the noise spectral density of the RF communication channel, $h_{k,l}$ is the gain of the lth sub-channel from sensor k, $0 < \phi < 1$ is a penalty factor due to practical digital coding, $\mu > 0$ is a factor due to analog waveform modulation, and $0 < \varphi < 1$ is a penalty factor due to the headings of data packets. A special case of the above energy model with $\varphi = 1$ is applied in [20], which studied the problem of minimizing MSE subject to energy constraint for single-hop sensor networks.

Suppose that MSE_0 is the desired MSE value at the destination node. Obviously, we can set up the same MSE constraint as in Algorithm 1

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \alpha_k \frac{W_m^2}{2^{2B_{k,m}}} \leq \eta \tag{3.36}$$

where $\eta = \frac{1}{4}(MSE_0 - \xi)$.

Given the above preparations, we can now formulate the following optimization problem to determine $B_{k,m}$ and $B'_{k,l}$ for all k, m and l.

$$\min_{B_{k,m},B'_{k,l}} J_p = \sum_{k=1}^{K-1}\sum_{l=1}^{L} a_{k,l}^p \left(2^{B'_{k,l}/\varphi} - 1\right)^p \tag{3.37}$$

subject to

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \leq \eta \tag{3.38}$$

$$\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l} \; for \; all \; k \tag{3.39}$$

$$B_{k,m} \geq 0, B'_{k,l} \geq 0 \; for \; all \; k \; , \; m, \; and \; l \tag{3.40}$$

where $J_p^{1/p}$ is the pth norm ($p \geq 1$) of all components of the energy cost in the network. If we choose $p = 1$, the cost corresponds to the sum energy. If we choose a large p, the cost corresponds approximately to the largest component of the energy cost. The problem (3.37) is more general than that formulated in [25] where $2^{B'_{k,l}/\varphi} - 1$ was replaced by its upper bound $2^{B'_{k,l}/\varphi}$, and $B'_{k,l}$ was chosen to be invariant to the sub-channel index l.

One can verify that if $B_{k,m}$ and $B'_{k,l}$ are treated as real numbers, the above problem is convex by checking the second order derivatives of objective function (3.37) and inequality constraint (3.38) as below.

$$\frac{\partial^2 J_p}{\partial {B'_{k,l}}^2} = \begin{cases} a_{k,l} 2^{B'_{k,l}/\varphi} \left(\frac{\ln 2}{\varphi}\right)^2, & p = 1 \\ a_{k,l}^p p \left(2^{B'_{k,l}/\varphi} - 1\right)^{p-2} 2^{B'_{k,l}/\varphi} \left(\frac{\ln 2}{\varphi}\right)^2 \left(p 2^{B'_{k,l}/\varphi} - 1\right), & p > 1 \end{cases} \tag{3.41}$$

The second order derivative of the left-side of inequality constraint (3.38), which we denote as a function $f(B_{k,m})$ is given by

$$\frac{\partial^2 f}{\partial B_{k,m}^2} = 4\alpha_k W_m^2 (\ln 2)^2 2^{-2B_{k,m}} \tag{3.42}$$

It is easy to see that $\frac{\partial^2 J_p}{\partial {B'_{k,l}}^2} \geq 0$ and $\frac{\partial^2 f}{\partial B_{k,m}^2} \geq 0$ for $p \geq 1$, $B_{k,m} \geq 0$ and $B'_{k,l} \geq 0$. The convex objective (3.37), convex inequality constraints (3.38) and (3.40), and the affine equality constraint (3.39), can guarantee that this optimization problem is a convex problem. Due to its convexity, the global optimal solution to the above problem can be found in principal. However, using a general-purpose program such as in Matlab, the computational speed is very slow. In the following, we present an efficient algorithm to solve this problem.

In this section, we develop an energy planning algorithm for progressive estimation by solving (3.37). To distinguish this algorithm from the previous algorithm 1 in section 3.6.1, we call it algorithm 2. To solve the problem (3.37), we apply the KKT method [56]. The complete set of the KKT equations for (3.37) are given by

$$a_{k,l}^p p \frac{\ln 2}{\varphi} 2^{B'_{k,l}/\varphi} \left(2^{B'_{k,l}/\varphi} - 1\right)^{p-1} - \nu_{k,l} + \xi_k = 0, \ for\ all\ k\ and\ l \tag{3.43}$$

$$-2\mu\alpha_k W_m^2 (\ln 2) 2^{-2B_{k,m}} - \lambda_{k,m} - \xi_k = 0, \ for\ all\ k\ and\ m \tag{3.44}$$

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} - \eta \leq 0 \tag{3.45}$$

$$\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l} \ for\ all\ k \tag{3.46}$$

$$\mu \left(\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} - \eta \right) = 0 \tag{3.47}$$

$$\lambda_{k,m} B_{k,m} = 0, \ for\ all\ k\ and\ m \tag{3.48}$$

$$\nu_{k,l} B'_{k,l} = 0, \ for\ all\ k\ and\ l \tag{3.49}$$

where $B_{k,m} \geq 0$, $B'_{k,l} \geq 0$, $\mu \geq 0$, $\lambda_{k,m} \geq 0$ and $\nu_{k,l} \geq 0$. It is easy to verify from the KKT equations that $\lambda_{k,m} = 0$ for all k and m where $B_{k,m} > 0$, and $\nu_{k,l} = 0$ for all k and l where $B'_{k,l} > 0$. It is also easy to verify that $\mu > 0$.

Finding the complete solution to the above nonlinear equations requires iterative search. To derive our search algorithm, we first define

$$\eta_k \doteq \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \tag{3.50}$$

From (3.44), we have that for $B_{k,m} > 0$,

$$\frac{\alpha_k W_m^2}{2^{2B_{k,m}}} = \frac{-\xi_k}{2\mu \ln 2} \tag{3.51}$$

Using (3.51) in (3.50) yields

$$\eta_k = \frac{-\xi_k |\mathcal{M}_k^+|}{2\mu \ln 2} + \sum_{m \notin \mathcal{M}_k^+} \alpha_k W_m^2 \tag{3.52}$$

where $\mathcal{M}_k^+ = \{m | B_{k,m} > 0\}$ and $|\mathcal{M}_k^+|$ is the size of the set $\mathcal{M}_k^+$. Substituting $\frac{\xi_k}{\mu}$ from (3.52) into (3.51), we find

$$B_{k,m} = \frac{1}{2} \log_2 \frac{\alpha_k W_m^2 |\mathcal{M}_k^+|}{\eta'_k} \tag{3.53}$$

where $m \in \mathcal{M}_k^+$, and $\eta'_k = \eta_k - \sum_{m \notin \mathcal{M}_k^+} \alpha_k W_m^2$. A slightly more compact form of this solution is

$$B_{k,m} = \left(\frac{1}{2} \log_2 \frac{\alpha_k W_m^2 |\mathcal{M}_k^+|}{\eta'_k} \right)^+ \tag{3.54}$$

where $(x)^+ = \max(x, 0)$.

To compute the solution (3.54) with a given η_k, we initially assume that $\mathcal{M}_k^+$ contains all $m = 1, \cdots, M$. We then apply (3.54) to calculate $B_{k,m}$ for all m. For those

$B_{k,m} = 0$, we exclude the corresponding indexes m from the set $\mathcal{M}_k^+$. The new set $\mathcal{M}_k^+$ is then applied to calculate new $B_{k,m}$ for all m via (3.54). This iterative procedure continues until $B_{k,m} > 0$ for $m \in \mathcal{M}_k^+$ and $B_{k,m} = 0$ for $m \notin \mathcal{M}_k^+$. In the appendix, we prove that the number of iterations in computing (3.54) is finite.

The first term in (3.43) is monotonically increasing function of $B'_{k,l} > 0$. So, for a given $B'_{k,l} > 0$, there is an unique and distinct ξ_k. Similarly, for a given ξ_k, there is either a unique $B'_{k,l} > 0$ or $B'_{k,l} = 0$. The computation of this one-to-one mapping can be easily implemented via a bisection method. For convenience, we can write $B'_{k,l} = f^+_{k,l}(\xi_k)$ where the function $f_{k,l}$ is the inverse function of the first term of (3.43). Note that we do not need any more explicit expression of $f_{k,l}$ for the reason mentioned already. Because of (3.46), we can find ξ_k by solving

$$\sum_{l=1}^{L} f^+_{k,l}(\xi_k) = B_k \tag{3.55}$$

where $B_k = \sum_{m=1}^{M} B_{k,m}$. The entire left-hand-side expression of (3.55) is a monotonically increasing function of ξ_k. So, with a given B_k, it is easy to find the corresponding ξ_k and hence the corresponding $B'_{k,l} = f^+_{k,l}(\xi_k)$ for all l.

Up to now, we have obtained the optimal solutions for $B_{k,m}$ and $B'_{k,l}$ for all k, m and l provided that η_k for all k are given. To find the optimal η_k, we need a search algorithm as developed next. Since $\mu > 0$, the optimal η_k must be such that $\sum_{k=1}^{K-1} \eta_k = \eta$ due to (3.50) and (3.47). Using $\sum_{k=1}^{K-1} \eta_k = \eta$ and (3.52), we can eliminate μ and obtain

$$\eta_k = \eta' \frac{\xi_k |\mathcal{M}_k^+|}{\sum_{j=1}^{K-1} \xi_j |\mathcal{M}_j^+|} + \sum_{m \notin \mathcal{M}_k^+} \alpha_k W_m^2 \tag{3.56}$$

where $\eta' = \eta - \sum_{j=1}^{K-1} \sum_{m \notin \mathcal{M}_j^+} \alpha_j W_m^2$. Here η_k is expressed as a function of $\mathcal{M}_j^+$ and ξ_j for

all j. Note that $\mathcal{M}_j^+$ is a by-product of $B_{j,m}$ for all m, and ξ_j is a by-product of $B'_{j,l}$ for all l.

The above discussions show that the computation of η_k for all k and the computation of $B_{k,m}$ and $B'_{k,l}$ for all k, m and l can be done alternately until convergence. To avoid divergence of the algorithm, we need to control the amount of changes in η_k for all k at each iteration. To do that, at iteration s, we introduce

$$\eta_k(s+1) = \beta\hat{\eta}_k(s+1) + (1-\beta)\eta_k(s) \tag{3.57}$$

where $1 \geq \beta > 0$, and $\hat{\eta}_k(s+1)$ is obtained from (3.56) at iteration $s+1$.

With the previous results, our algorithm for computing the solution to the problem (3.37) can be summarized below:

1. Initialization: Choose $\mathcal{M}_k^+ = \{1, 2, \cdots, M\}$ for all k. Also set $s = 0$ and $\eta_k(0) = \frac{\eta}{K-1}$ for all k.

2. Step 1: Compute $B_{k,m}$ and $\mathcal{M}_k^+$ by (3.54).

3. Step 2: Compute $B'_{k,l}$ and ξ_k by (3.55).

4. Step 3: Compute $\eta_k(s+1)$ by (3.57).

5. Step 4: Set $s := s+1$, and go to Step 1 until convergence.

The simulations show that for $p = 1$, this algorithm converges for all $\beta \in (0, 1]$, but for $p > 1$, $\beta < 1$ is necessary for the algorithm to converge. In the simulation examples shown later, we used $\beta = 0.02$ for all cases where $p > 1$. Note that for $p = 1$, the first term in

(3.43) is nonzero even if $B'_{k,l} = 0$, but for $p > 1$, this term equals zero when $B'_{k,l} = 0$. Also shown in simulation, all other random initializations of η_k converged to the same result.

After the solution converges, in practice, we round up the number of quantization bits $B_{k,m}$ to the nearest integer, i.e., $\bar{B}_{k,m} = \lceil B_{k,m} \rceil$. Because coding can be done jointly over multiple sub-channels, $B'_{k,l}$, the number of bits per symbol transmitted per channel, could be real number in principle. We do not round up $B'_{k,l}$. But we need to apply (3.55) one more time to calculate the final $B'_{k,l}$ with $B_k = \sum_{m=1}^{M} \lceil B_{k,m} \rceil$.

3.6.3 Linear Energy Model for Progressive Estimation

Suppose that there is a large number of sub-channels for each transmission and the gains of the sub-channels are the same. Then, the optimal $B'_{k,l}$ should be very small for each k and l, and hence (3.35) becomes $E_{k,l} = a_{k,l} \frac{\ln 2}{\varphi} B'_{k,l}$ which is a linear energy model.

Under the linear model, if $a_{k,l}$ is not invariant to l, then the optimal $B'_{k,l}$ would be such that only the largest $a_{k,l}$ over all l is allocated with non-zero $B'_{k,l}$ which could violate the assumption that all $B'_{k,l}$ are small subject to $\sum_{l=1}^{L} B'_{k,l} = \sum_{m=1}^{M} B_{k,m} = B_k$. However, if we simply force $B'_{k,l} = B_k/L$ which is invariant to l, then the linear model implies $\sum_{l=1}^{L} E_{k,l} = \frac{1}{L} \left(\sum_{l=1}^{L} a_{k,l} \frac{\ln 2}{\varphi} \right) B_k$.

The linear energy model also applies to many existing communication devices where the energy cost is simply proportional to the number of packets transmitted. Therefore, there is a need to consider this special case. We will write

$$E_k = b_k B_k \tag{3.58}$$

where E_k is the energy spent by sensor k to transmit B_k bits, and b_k is a constant associated

with sensor k. For comparison with the algorithm in Section 3.6.2, we will choose $b_k = \frac{1}{L}\sum_{l=1}^{L} a_{k,l}\frac{\ln 2}{\varphi}$ in the simulation section. To minimize the sum energy by forcing $B'_{k,l} = B_k/L$ under the linear energy model, we need to solve:

$$\min_{B_{k,m}} \sum_{k=1}^{K-1}\sum_{m=1}^{M} b_k B_{k,m} \tag{3.59}$$

subject to

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \leq \eta \tag{3.60}$$

$$B_{k,m} \geq 0, \ for\ all\ k\ and\ m \tag{3.61}$$

This problem could be reduced from (3.37) if $p = 1$, $a_{k,l}$ is invariant to l, and $B'_{k,l} << 1$. However, this problem formulation stands on its own without the above conditions on $a_{k,l}$ and $B'_{k,l}$. We also like to mention that for the linear energy model, the basic energy component is E_k. It is hence tempting to minimize the more general cost $\sum_{k=1}^{K-1} E_k^p$ with any $p \geq 1$. For large p, the cost automatically weighs more for sensors that consume more energy. But unfortunately, the resulting optimization problem with respect to the variables $B_{k,m}$ is much more difficult, which will not be addressed in this thesis. Next, we present a simple algorithm to solve (3.59).

The KKT conditions of this problem are given by

$$b_k - 2(\ln 2)\mu\alpha_k W_m^2 2^{-2B_{k,m}} - \lambda_{k,m} = 0, \ for\ all\ k\ and\ m \tag{3.62}$$

$$\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} \leq \eta \tag{3.63}$$

$$\mu\left(\sum_{k=1}^{K-1}\sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} - \eta\right) = 0 \tag{3.64}$$

$$\lambda_{k,m} B_{k,m} = 0, \ for\ all\ k\ and\ m \tag{3.65}$$

where $\mu \geq 0$, $\lambda_{k,m} \geq 0$ and $B_{k,m} \geq 0$. We define the set $\mathcal{KM}^+$ containing the indices of (k,m) where $B_{k,m} > 0$. For $(k,m) \notin \mathcal{KM}^+$, $B_{k,m} = 0$. From (3.65), we can see $\lambda_{k,m} = 0$ where $(k,m) \in \mathcal{KM}^+$. Plugging this result into (3.62), we can see $\mu > 0$. We then see from (3.64) that

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\alpha_k W_m^2}{2^{2B_{k,m}}} = \eta \tag{3.66}$$

For $(k,m) \in \mathcal{KM}^+$, we know that $\lambda_{k,m} = 0$ and hence from (3.62) that

$$B_{k,m} = \left(\frac{1}{2} \log \frac{2(\ln 2)\mu\alpha_k W_m^2}{b_k} \right)^+ \tag{3.67}$$

where the constant μ is chosen such that (3.66) holds.

3.7 Progressive Estimation with Analog Transmission

We now consider progressive estimation with analog transmissions between sensors. Recall the analog transmission model discussed in Section 3.2 and Appendix 3.11.1. We will assume M identical discrete-time analog sub-channels between each sensor and its downstream sensor and the M sub-channels are used to transmit in parallel the M elements of $\hat{\boldsymbol{\theta}}_k$ from sensor k to its downstream sensor. Applying (3.91) and (3.92), we can formulate an effective channel model for analog transmission between any two sensors as follows: the received symbol at the downstream counterpart of sensor k in the mth sub-channel is

$$y_{k,m} = h_{k,m} \sqrt{\frac{E_{c_{k,m}}}{N_0}} \hat{\theta}_{k,m} + \nu_{k,m} \tag{3.68}$$

where $E_{c_{k,m}}$ is the energy of the waveform used in modulation for the mth element $\hat{\theta}_{k,m}$ of the estimate $\hat{\boldsymbol{\theta}}_k$ at sensor k, and the noise $\nu_{k,m}$ is a complex Gaussian random variable with

zero mean and unit variance. The transmission energy for the mth element is $|\hat{\theta}_{k,m}|^2 E_{c_{k,m}}$. Then, by combining all M sub-channels, we have the vector channel model:

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{A}_k \hat{\boldsymbol{\theta}}_k + \boldsymbol{\nu}_k \tag{3.69}$$

where $\boldsymbol{A}_k = diag\left[\sqrt{\frac{E_{c_{k,1}}}{N_0}}, \sqrt{\frac{E_{c_{k,2}}}{N_0}}, \cdots, \sqrt{\frac{E_{c_{k,M}}}{N_0}}\right]$ and $\boldsymbol{H}_k = diag[h_{k,1}, h_{k,2}, \cdots, h_{k,M}]$. For the analog transmission, we allow the channel gains of the sub-channels to be possibly different from each other. Denoted by P_k, the total transmission energy for transmitting $\hat{\boldsymbol{\theta}}_k$ from sensor k to its downstream sensor using the analog mode is now $P_k = N_0 Tr\{\boldsymbol{A}_k \hat{\boldsymbol{\theta}}_k \hat{\boldsymbol{\theta}}_k^H \boldsymbol{A}_k\}$. The energy planning for analog transmission is about the design of $\boldsymbol{A}_k$ for all k.

3.7.1 BLUE

Based on the local observation $\boldsymbol{x}_k$ at sensor k and the data $\boldsymbol{y}_l$ for $l \in S_k$ received by sensor k from its upstream sensors, the BLUE of $\boldsymbol{\theta}$ at sensor k is

$$\hat{\boldsymbol{\theta}}_k = \left(\boldsymbol{G}_k^H \boldsymbol{G}_k + \sum_{l \in S_k} \boldsymbol{B}_l^H \boldsymbol{C}_{\boldsymbol{y}_l}^{-1} \boldsymbol{B}_l\right)^{-1} \left(\boldsymbol{G}_k^H \boldsymbol{x}_k + \sum_{l \in S_k} \boldsymbol{B}_l^H \boldsymbol{C}_{\boldsymbol{y}_l}^{-1} \boldsymbol{y}_l\right) \tag{3.70}$$

where $\boldsymbol{B}_l = \boldsymbol{H}_l \boldsymbol{A}_l$, $\boldsymbol{C}_{\boldsymbol{y}_l} = \boldsymbol{B}_l \boldsymbol{C}_l \boldsymbol{B}_l^H + \boldsymbol{I}$, and $\boldsymbol{C}_l$ is the covariance of $\hat{\boldsymbol{\theta}}_l$. The covariance matrix of $\hat{\boldsymbol{\theta}}_k$ is given by

$$\boldsymbol{C}_k = \left(\boldsymbol{G}_k^H \boldsymbol{G}_k + \sum_{l \in S_k} \boldsymbol{B}_l^H \left(\boldsymbol{B}_l \boldsymbol{C}_l \boldsymbol{B}_l^H + \boldsymbol{I}\right)^{-1} \boldsymbol{B}_l\right)^{-1} \tag{3.71}$$

Based on (3.71), the MSE at the destination node is a very complicated function of the amplification matrices $\boldsymbol{A}_k$, $k = 1, \cdots, K$. It is not feasible to use (3.71) to design energy planning unless the network is very small.

3.7.2 Averaging

Alternatively, at sensor k, we can obtain an estimate of $\boldsymbol{\theta}_k$ by averaging the BLUEs of $\boldsymbol{\theta}$ based on $\boldsymbol{y}_l$ for $l \in S_k$ and $\boldsymbol{x}_k$ individually, i.e.,

$$\tilde{\boldsymbol{\theta}}_k = \frac{1}{I_k+1}\left(\boldsymbol{G}_k^H\boldsymbol{G}_k\right)^{-1}\boldsymbol{G}_k^H\boldsymbol{x}_k + \frac{1}{I_k+1}\sum_{l\in S_k}\boldsymbol{B}_l^{-1}\boldsymbol{y}_l \tag{3.72}$$

where $\boldsymbol{B}_l$ for $l \in S_k$ are needed at sensor k. Denote the covariance matrix of $\tilde{\boldsymbol{\theta}}_k$ by $\tilde{\boldsymbol{C}}_k$. Then, the covariance matrix of $\boldsymbol{y}_l$ is $\boldsymbol{B}_l\tilde{\boldsymbol{C}}_l\boldsymbol{B}_l^H + \boldsymbol{I}$, and the covariance matrix of $\tilde{\boldsymbol{\theta}}_k$ is given by

$$\tilde{\boldsymbol{C}}_k = \frac{1}{(1+I_k)^2}\left(\left(\boldsymbol{G}_k^H\boldsymbol{G}_k\right)^{-1} + \sum_{l\in S_k}\left(\boldsymbol{B}_l^H\boldsymbol{B}_l\right)^{-1} + \sum_{l\in S_k}\tilde{\boldsymbol{C}}_l\right) \tag{3.73}$$

The recursion (3.73) is linear and much simpler than the recursion (3.71). We will use (3.73) for energy planning. It follows from Lemma 1 that $\tilde{\boldsymbol{C}}_k \geq \boldsymbol{C}_k$.

3.8 Energy Planning with Analog Transmission

Using the recursion (3.73), the covariance matrix at the destination node, labelled as node K, can be found as

$$\tilde{\boldsymbol{C}}_K = \sum_{k=1}^{K-1}\alpha_k\left(\boldsymbol{B}_k^H\boldsymbol{B}_k\right)^{-1} + \sum_{k=1}^{K}\frac{\alpha_k}{(1+I_k)^2}\left(\boldsymbol{G}_k^H\boldsymbol{G}_k\right)^{-1} \tag{3.74}$$

where $\alpha_K = 1$ and $\alpha_l = \frac{\alpha_k}{(1+I_k)^2}$ for $l \in S_k$. Then, the MSE at the destination node is

$$MSE_K = Tr(\tilde{\boldsymbol{C}}_K) = \sum_{k=1}^{K-1}\sum_{m=1}^{M}\frac{\beta_{k,m}^2}{a_{k,m}^2} + \epsilon \tag{3.75}$$

where $\epsilon = Tr\left(\sum_{k=1}^{K}\frac{\alpha_k}{(1+I_k)^2}\left(\boldsymbol{G}_k^H\boldsymbol{G}_k\right)^{-1}\right)$ and $\beta_{k,m} = \frac{\sqrt{\alpha_k}}{|h_{k,m}|}$. Here, $h_{k,m}$ is the mth diagonal element of $\boldsymbol{H}_k$, and $a_{k,m} = \sqrt{E_{c_{k,m}}/N_0}$ is the mth diagonal element of $\boldsymbol{A}_k$. Then it follows

that

$$P_k = N_0 Tr\left(\boldsymbol{A}_k \tilde{\boldsymbol{\theta}}_k \tilde{\boldsymbol{\theta}}_k^H \boldsymbol{A}_k\right) \leq \sum_{m=1}^{M} N_0 a_{k,m}^2 2W_m^2 = \sum_{m=1}^{M} \bar{P}_{k,m} \tag{3.76}$$

where $\bar{P}_{k,m} = N_0 a_{k,m}^2 2W_m^2$, which can be viewed as the upper bound of the energy used to transmit the mth element of $\tilde{\boldsymbol{\theta}}_k$. The inequality comes with the assumption that both the real and image part of $\hat{\theta}_{k,m}$ or $\tilde{\theta}_{k,m}$ are bounded within $[-W_m, W_m]$, which is consistent with the digital transmission case.

Similar to the digital transmission energy planning, we define the following cost function for the analog transmission energy planning:

$$F_p = \sum_{k=1}^{K-1} \sum_{m=1}^{M} \bar{P}_{k,m}^p = \sum_{k=1}^{K-1} \sum_{m=1}^{M} a_{k,m}^{2p} (\sqrt{2N_0} W_m)^{2p} \tag{3.77}$$

which is the pth power of the L_p norm of $\bar{P}_{k,m}$ over all k and m. And the analog transmission energy planning is formulated as follows:

$$\min_{a_{k,m}} F_p \tag{3.78}$$

subject to the MSE constraint at node K:

$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} \frac{\beta_{k,m}^2}{a_{k,m}^2} \leq MSE_0 - \epsilon \triangleq \eta \tag{3.79}$$

We now define $A_{k,m} = (a_{k,m}^{2p} (\sqrt{2N_0} W_m)^{2p})^{1/(p+1)}$ and $C_{k,m} = (\beta_{k,m}^2 / a_{k,m}^2)^{p/(p+1)}$. Following the Hölder's inequality, we have

$$\left(\sum_{k=1}^{K-1} \sum_{m=1}^{M} A_{k,m}^{(p+1)}\right)^{1/(p+1)} \left(\sum_{k=1}^{K-1} \sum_{m=1}^{M} C_{k,m}^{(p+1)/p}\right)^{p/(p+1)} \geq$$
$$\sum_{k=1}^{K-1} \sum_{m=1}^{M} A_{k,m} C_{k,m} = (\sqrt{2N_0} W_m)^{2p/(p+1)} \beta_{k,m}^{2p/(p+1)} \triangleq \gamma \tag{3.80}$$

where equality holds when there is a constant λ such that

$$A_{k,m}^{p} = \lambda C_{k,m} \tag{3.81}$$

Combining (3.79) and (3.80), we have

$$F_p = \left(\sum_{k=1}^{K-1} \sum_{m=1}^{M} A_{k,m}^{(p+1)} \right) \geq \frac{\gamma^{p+1}}{\eta^p} \tag{3.82}$$

The right-hand-side of (3.82) is independent of $a_{k,m}$ and is the minimum of F_p. This minimum is achieved if the equality in (3.79) is achieved and the equality (3.81) holds. From (3.81), one can easily verify that

$$a_{k,m} = \lambda^{\frac{1}{2p}} \beta_{k,m}^{\frac{1}{p+1}} (\sqrt{2N_0} W_m)^{-\frac{p}{p+1}} \tag{3.83}$$

Applying this to the equality in (3.79) yields

$$\lambda = \left(\frac{\sum_{k=1}^{K-1} \sum_{m=1}^{M} (\sqrt{2N_0} W_m)^{\frac{2p}{p+1}} \beta_{k,m}^{\frac{2p}{p+1}}}{\eta} \right)^{p} \tag{3.84}$$

Combing (3.83) and (3.84), we finally get

$$a_{k,m} = \left(\beta_{k,m}^{\frac{1}{p+1}} (\sqrt{2N_0} W_m)^{-\frac{p}{p+1}} \right) \sqrt{\frac{\sum_{k=1}^{K-1} \sum_{m=1}^{M} (\sqrt{2N_0} W_m)^{\frac{2p}{p+1}} \beta_{k,m}^{\frac{2p}{p+1}}}{\eta}} \tag{3.85}$$

Similar to the digital case, the energy planning here is based on the L_p norm for any p. If $p \to \infty$, (3.85) can be rewritten as follows:

$$a_{k,m} = (\sqrt{2N_0} W_m)^{-1} \sqrt{\lambda'''} \tag{3.86}$$

where $\lambda''' = \frac{\sum_{k=1}^{K-1} \sum_{m=1}^{M} (\sqrt{2N_0} W_m)^2 \beta_{k,m}^2}{\eta}$, and hence $\bar{P}_{k,m} = N_0 a_{k,m}^2 2W_m^2 = \lambda'''$, which is independent of k and m.

3.9 Simulation Results

The simulation network we consider is shown in Fig. 3.2 where there are $K = 400$ nodes. The destination node is also referred to as fusion center. This network was constructed in such a way that the distance between a sensor and its upper stream sensor is $D\delta$ where δ is uniformly distributed within the range [0.5, 1.5] and D is a normalizing factor.

For the simulation, we assume that sensor k observes the data vector $\boldsymbol{x}_k = \boldsymbol{G}_k\boldsymbol{\theta} + \boldsymbol{\omega}_k$, where $\boldsymbol{G}_k$ is the observation matrix associated with sensor k, which is assumed known to sensor k, and $\boldsymbol{\omega}_k$ is white noise with the identity covariance matrix $\boldsymbol{C}_\omega = \boldsymbol{I}$. With this observation model, an original local estimate of $\boldsymbol{\theta}$ at sensor k can be obtained by the best linear unbiased estimate: $\boldsymbol{\theta}_k = (\boldsymbol{G}_k^T\boldsymbol{G}_k)^{-1}\boldsymbol{G}_k^T\boldsymbol{x}_k$. We also have $\boldsymbol{C}_{\theta,k} = (\boldsymbol{G}_k^T\boldsymbol{G}_k)^{-1}$. We will choose $\boldsymbol{G}_k$ for $k = 1, 2, \cdots, K$ randomly. Each $\boldsymbol{G}_k$ is a $N \times M$ real matrix with elements randomly chosen from a Gaussian distribution with zero mean and standard deviation equals to 10.

Each entry of $\boldsymbol{\theta}$ is chosen randomly from $[-1, 1]$. The squared channel gain of channel l from sensor k to its downstream sensor is $|h_{k,l}|^2 = d_k^{-\alpha}\rho_{k,l}$, where d_k is the distance from sensor k to its down-stream sensor, $\alpha = 4$ and $\rho_{k,l}$ is randomly chosen from an exponential distribution with mean equal to one. We also choose $N = 20$, $M = 10$, $\mu = 1$, $\phi = 1$, $N_0 = 1$, $W_m = 1$ for $m = 1, 2, \cdots, M$.

Before we show the energy cost of all energy planning algorithms, please recall that for digital transmissions energy planning algorithm 1, the energy consumed by sensor k is denoted by E_k in (3.15) and the p-norm cost function is denoted by J_p in (3.17). For digital

transmissions energy planning algorithm 2, the energy consumed by sensor k is denoted by $E_k = \sum_{l=1}^{L} E_{k,l}$ where $E_{k,l}$ is given in (3.35) and the p-norm cost function is denoted by J_p in (3.37). For analog transmissions, the energy consumed by sensor k is denoted by P_k in (3.76) and the p-norm cost function is denoted by F_p in (3.77).

3.9.1 Analog Transmission Energy Planning

The energy planning algorithm present in section 3.8 for progressive estimation with analog transmissions will be referred to as analog proposed progressive (PP) energy planning algorithm for multihop sensor network with analog transmission. For comparison, we also consider a uniform progressive (UP) energy planning algorithm for the same network. For the UP algorithm, each sensor uses the same transmission matrix $\boldsymbol{A}_k = a\boldsymbol{I}$ where a is chosen to achieve the target MSE at the destination node, i.e., $MSE_K \leq MSE_0$. In simulations, once the transmission matrices $\boldsymbol{A}_k$, $k = 1, \cdots, K$, are determined, we use the second term in (3.76) to calculate the transmission energy at sensor k.

Using the cost F_1, Fig. 3.3 illustrates the total transmission energy consumed by the entire network (averaged over 100 realizations of $\boldsymbol{\theta}$) versus the target MSE denoted by MSE_0. We see that the PP algorithm consumes much less total energy than the UP algorithm. We also see that for either the UP algorithm or the PP algorithm, the BLUE estimation algorithm consumes less energy than the averaging estimation algorithm. To illustrate the advantage of the multi-hop network over the single-hop network in terms of the transmission energy consumption, Fig. 3.3 also shows a curve of the energy versus the target MSE based on a single-hop tree for the same distribution of sensors shown in Fig. 3.2.

We see an enormously large gap of energy between the single-hop tree and the multi-hop tree. The path loss exponent is assumed to be four.

For $MSE_0 = 0.05$, Fig. 3.4 illustrates the transmission energy consumed by each sensor versus the normalized distance (i.e., the distance divided by D) of the sensor from the destination node. A single random realization of $\boldsymbol{\theta}$ is used. For this figure, the PP algorithm uses F_1, and the sensors close to the fusion center needs to transmit much more energy than the sensors far away from the fusion center. But if we choose a large p (say, $p = 50$) for F_p, the PP algorithm can yield a virtually constant energy distribution and become virtually the same as the UP algorithm.

Fig. 3.5 shows the effect of p on the analog transmission energy planning. We can see that as p in the cost F_p increases, the total transmission energy (corresponding to F_1) increases. We also see as expected that when $p = \infty$, the PP algorithm becomes identical to the UP algorithm. See (3.86).

Fig. 3.6 illustrates the actual MSE MSE_K (averaged over 100 realizations of $\boldsymbol{\theta}$) at the destination node versus MSE_0. For the PP algorithm (using F_1), the averaging algorithm always achieves the target MSE at the destination, but the BLUE algorithm yields a much smaller MSE than the target MSE. This is because the covariance matrix of the estimate by averaging is not a tight upper bound of the covariance matrix of the estimate by the BLUE. In other words, the energy planning algorithm based on the averaging progressive estimation is rather conservative for the BLUE progressive estimation.

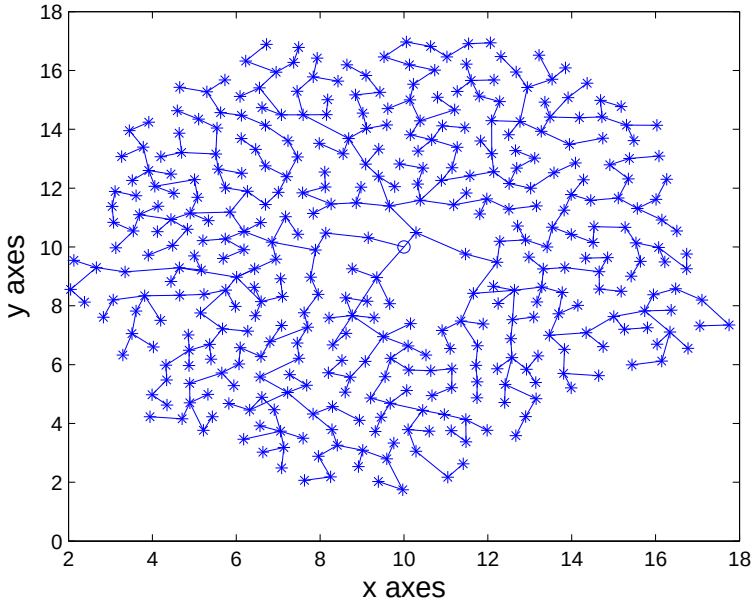

Figure 3.2: A 2-D multihop sensor network with a routing tree. The destination node (also referred to as the fusion center) is marked by the circle in the center. Here, there are 400 sensors, each marked by $*$. This network is used for all simulation examples in this thesis.

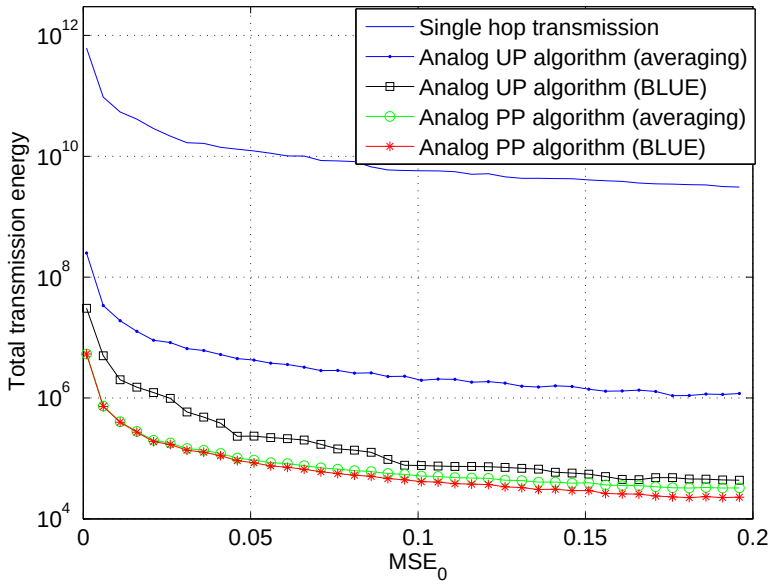

Figure 3.3: For the analog PP and UP algorithms, the transmission energy consumed by the network versus the target MSE_0. Note that the first curve on the top is based on a single-hop tree for the same distribution of sensors shown in Fig. 3.2. The path loss exponent is assumed to be four.

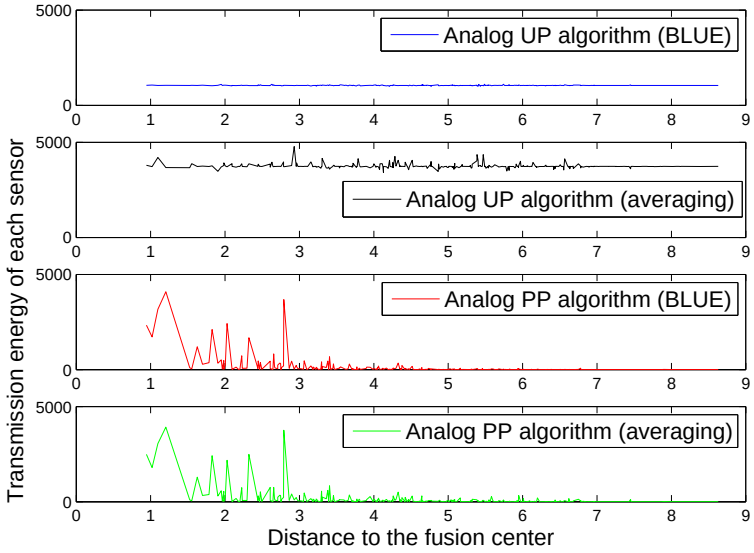

Figure 3.4: For the analog PP and UP algorithms, the transmission energy consumed by each sensor versus the normalized Euclidean distance between the sensor and the fusion center. $MSE_0 = 0.05$. The analog PP algorithm uses F_1.

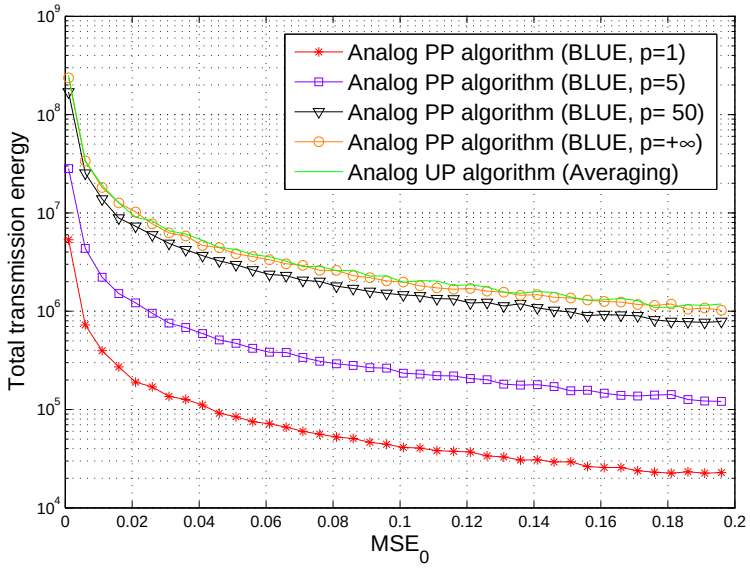

Figure 3.5: Illustration of the effect of p in F_p for the analog PP algorithm.

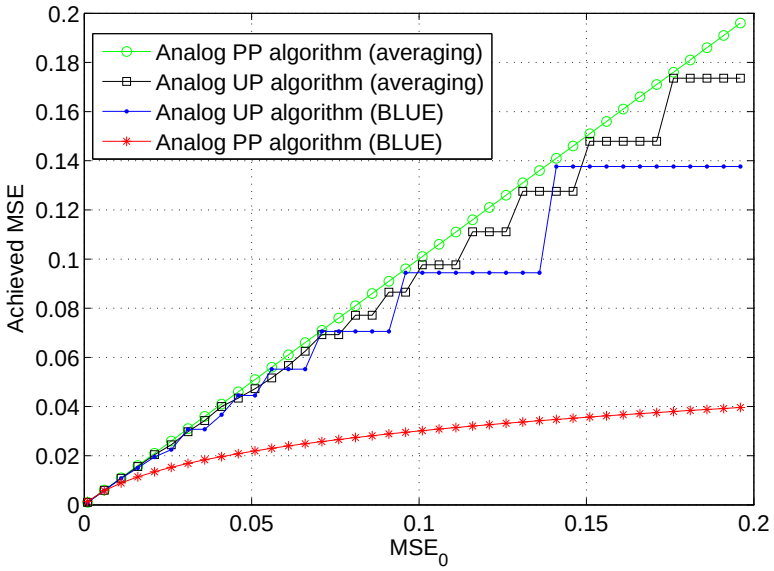

Figure 3.6: For the analog PP and UP algorithms, the actual MSE at the fusion center versus the target MSE.

3.9.2 Digital Transmission Energy Planning - Algorithm 1

We will compare the proposed progressive (PP) digital transmission energy planning algorithm 1 and a uniform progressive (UP) digital transmission energy planning algorithm. The digital UP algorithm here is such that the number of bits allocated to each element of the estimate by each sensor is a constant subject to the MSE constraint $MSE_K \leq MSE_0$ at the destination node.

For $L = M$, Fig. 3.7 illustrates the total normalized transmission energy consumed by the network, i.e., $E_{net} = \sum_{k=1}^{K} E_k/D^{\alpha}$, versus the target MSE. The energy E_k is computed based on (3.15). We see that the PP algorithm with small p consumes much less energy than the UP algorithm. But with a large p, the result of the PP algorithm becomes similar to, but not exactly the same as, that of the UP algorithm.

Under $MSE_0 = 1.0 \times 10^{-3}$, $L = M$ and J_1, Fig. 3.8 illustrates the number of bits per element for each sensor, i.e., $\sum_{m=1}^{M} B_{k,m}/M$, versus the Euclidean distance (divided by D) from the sensor to the destination node. We see that under the PP algorithm with J_1, the number of bits allocated to each sensor generally decreases with the distance from the sensor to the destination node.

Under the same condition as for Fig. 3.8, Fig. 3.9 shows the amount of the normalized transmission energy E_k/D^r consumed by each sensor versus the distance (divided by D) from the sensor to the destination node. Note that unlike the analog case, the transmission energy determined for each sensor for the digital case is not affected by the estimation algorithm (quasi BLUE or averaging).

Under $MSE_0 = 1.0 \times 10^{-3}$ and $L = M$, Fig. 3.10 shows the effect of a large

p on the bit distribution of the digital PP algorithm. And Fig. 3.11 shows the effect of a large p on the energy distribution of the digital PP algorithm. We see that the digital PP algorithm with a large p is more effective than the digital UP algorithm to achieve a constant energy distribution. The digital UP algorithm can not achieve a constant energy distribution because of the varying channel gains from sensor to sensor.

We now evaluate the effect of the transmission time-bandwidth product L. Under $MSE_0 = 1.0 \times 10^{-3}$, Fig. 3.12 shows the total transmission energy of the network versus the ratio L/M. We see that as L increases, less energy is consumed by any of the two algorithms (PP or UP). This figure suggests that if the network has a large bandwidth, we should use a large L to save transmission energy. However, we also see from this figure that the required transmission energy has a nonzero lower bound as L becomes very large, which is proved by (3.30). For the example shown in Fig. 3.12, $L = 4M$ is practically large enough to approach the bound.

Fig. 3.13 shows the actual MSE value at the destination node versus the target MSE value at the destination node. Such a curve depends on both the energy planning algorithm and the estimation algorithm. When the averaging algorithm is used, the actual MSE is quite close to the target MSE, which is expected. However, when the quasi-BLUE algorithm is used, the actual MSE is much smaller than the target MSE. One should recall that the quasi-BLUE algorithm requires each sensor to know the upper bounds of the covariance matrices of the quantized estimates from its upper stream sensors while the averaging algorithm does not have this requirement. Yet, given the large gap of the MSE between the two estimation algorithms as shown in this figure, developing a more efficient bit

allocation algorithm for the quasi-BLUE estimation algorithm remains a useful challenge.

3.9.3 Digital Transmission Energy Planning - Algorithm 2

The results of the energy planning algorithm depend on the choice of p in (3.37). However, regardless of the choice of p, the sum energy is given by

$$J_1 = \sum_{k=1}^{K-1} \sum_{l=1}^{L} a_{k,l} \left(2^{B'_{k,l}} - 1 \right) \tag{3.87}$$

The sum of transmission energy to be illustrated is given by J_1/D^{α}.

For convenience of reference, we will refer to the algorithm developed in Section 3.6.1 as the digital proposed progressive (PP) algorithm 1, the algorithm shown in Section 3.6.2 as the digital proposed progressive (PP) algorithm 2, and the algorithm under linear energy model in Section 3.6.3 as the digital linear progressive (LP) algorithm. We also have a digital "uniform" progressive (UP) algorithm for which the same number of quantization bits is assigned to each element of the estimate at each sensor (i.e., $B_{k,m}$ is independent of k and m), the same number of transmission bits is assigned to each sub-channel (i.e., $B'_{k,l}$ is independent of k and l), and however $\sum_{m=1}^{M} B_{k,m} = \sum_{l=1}^{L} B'_{k,l}$ and the MSE constraint (3.19) hold.

We like to note here that for $K = 40$, $N = 4$, $M = 2$, $L = 2$, $p = 1$, it took 2.3125 seconds on a computer (with Pentium (R) 4 CPU 3.00 GHz and 1G memory) for the digital PP algorithm 2 to find the solution of (3.37). But with the Matlab nonlinear constrained optimization routine *fmincon*, it took 219.56 seconds although the solution is the same. For the following example where $K = 400$, $N = 20$, $M = 10$, $L = 10$, and $p = 1$, the Matlab routine virtually froze up the computer.

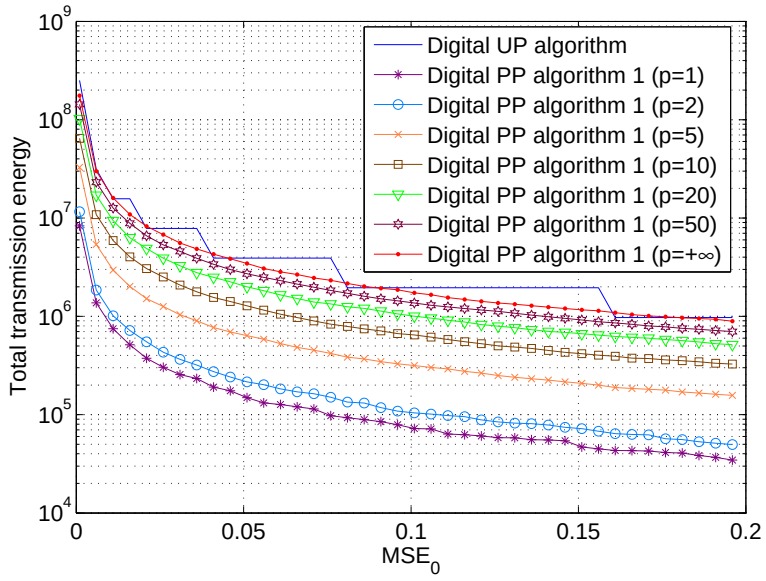

Figure 3.7: For the digital PP and UP algorithms, the transmission energy consumed by the network versus the target MSE where p is as in J_p for the digital PP algorithm.

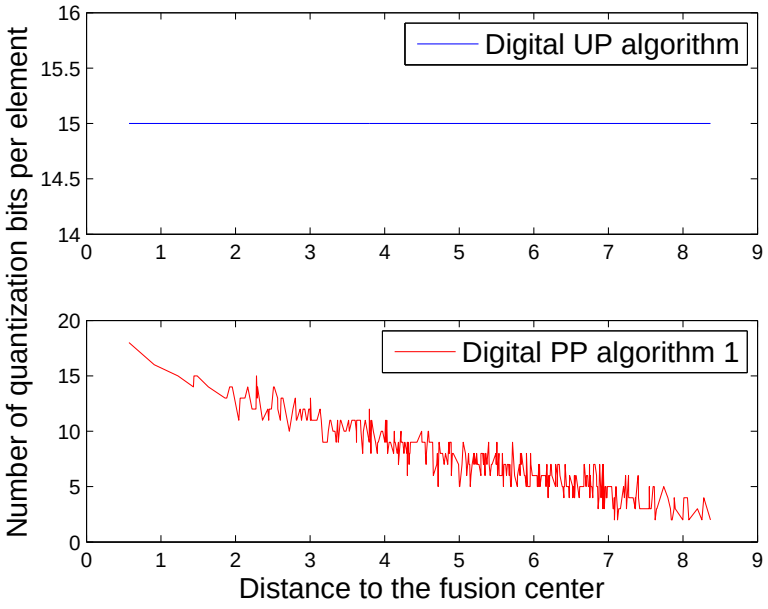

Figure 3.8: The number of quantization bits allocated for each sensor per element of the unknown vector versus the normalized Euclidean distance from the sensor to the destination node. The target MSE at the destination node is $MSE_0 = 1.0 \times 10^{-3}$. The digital PP algorithm uses J_1.

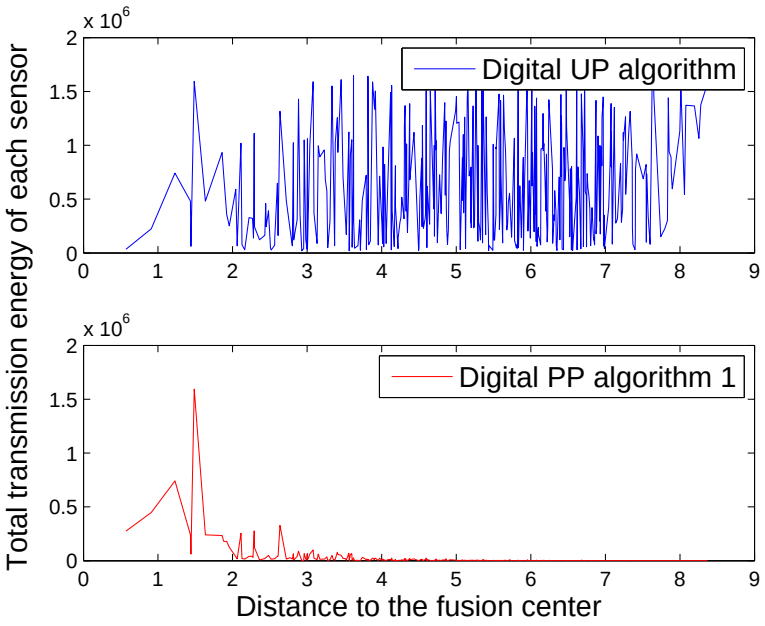

Figure 3.9: Normalized digital transmission energy by each sensor versus the normalized Euclidean distance from the sensor to the destination node. $MSE_0 = 1.0 \times 10^{-3}$. The digital PP algorithm uses J_1.

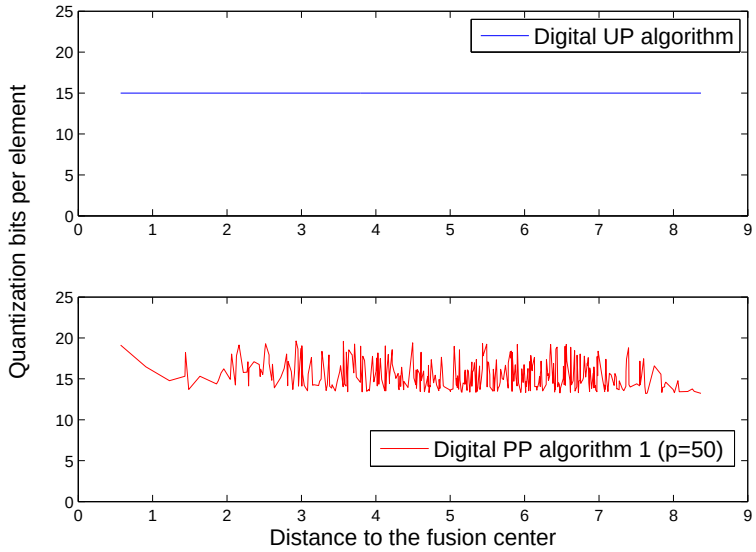

Figure 3.10: Same as Figure 3.8 except $p = 50$ for the digital PP algorithm.

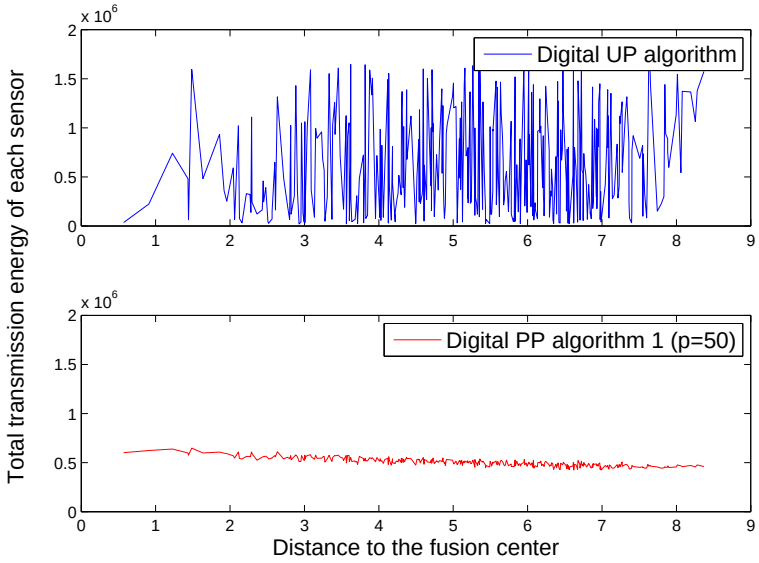

Figure 3.11: Same as Figure 3.9 except $p = 50$ for the digital PP algorithm.

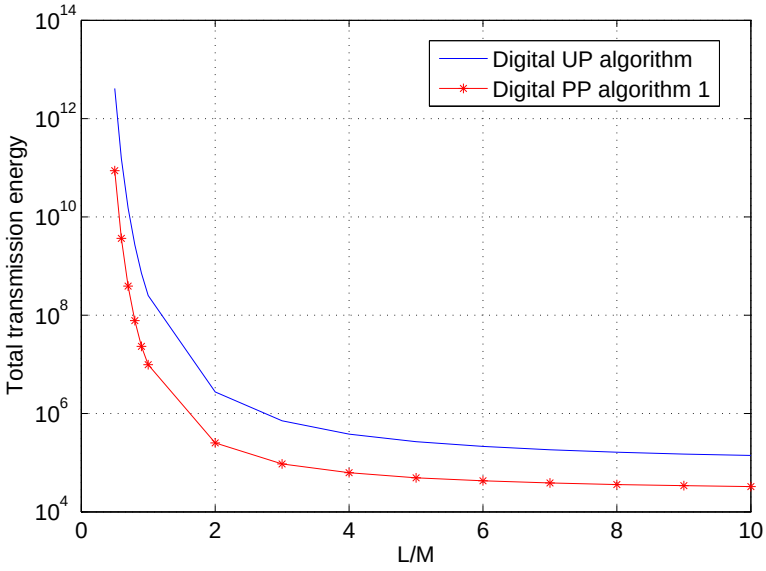

Figure 3.12: Total digital transmission energy consumed by the network versus L/M with $M = 10$. $MSE_0 = 1.0 \times 10^{-3}$.

Comparison of sum energy

Fig. 3.14 compares several curves of the sum energy versus the target MSE. Each of these curves (except one) is determined by one of the following energy planning algorithms for progressive estimation: digital PP algorithm 1 and 2 with $p = 1$, $p = 8$ and $p = 64$, , the digital LP algorithm, and the digital UP algorithm.

We see that with increased p, the sum energy determined by the digital PP algorithm 2 increases as expected. The same is true for the digital PP algorithm 1. However, the sum energy by the digital PP algorithm 2 is always smaller than that by the digital PP algorithm 1 for each given p. This is because algorithm 2 uses the exact energy model (as opposed to its upper bound) and also exploits the variation of the sub-channel gains.

We also see that the digital LP algorithm requires more sum energy than the digital PP algorithm 1 with $p = 1$, which is expected. Note that although the energy model used for developing the digital LP algorithm is linear, i.e., (3.58), the actual amount of energy required by all algorithms (as shown in all figures) is computed by using the original energy model (3.35) where for the digital LP algorithm we use $B'_{k,l} = B_k/L$. But when the number of sub-channels L becomes large, the digital LP algorithm and the digital PP algorithm 1 with $p = 1$ should require the same sum energy, which will be illustrated in the next figure.

The digital UP algorithm is clearly a bad choice for progressive estimation in terms of the sum energy cost.

Effect of the number of sub-channels

Fig. 3.15 illustrates the sum energy versus the number L of sub-channels, where the target MSE is 0.01. For all algorithms, the sum energy cost decreases as L increases, and becomes less sensitive to L when L is large. An explanation of this is available in [25]. As expected, the sum energy required by the digital LP algorithm becomes the same as that by the digital PP algorithm 1 when L is large. In this figure, the curves for the digital PP algorithm 1 and the digital PP algorithm 2 appear overlapping in the region of small L because of the large scale. In fact, the digital PP algorithm 2 is always better than the digital PP algorithm 1.

Effect of p on energy distribution

Fig. 3.16 shows the effect of p used in the digital PP algorithm 2 for progressive estimation on the peak energy consumed by individual sensors. We see that the peak energy is reduced when p is increased.

3.9.4 Analog versus Digital transmissions

To compare the analog case with the digital case, we consider the same network topology (Fig. 3.2), the same observation model (2.1) and the same RF channel with the same $\mathcal{W}$ the same $\mathcal{T}$. Fig. 3.17 compares the energy consumptions between the analog PP algorithm and the digital PP algorithm. We see that when $L = M$, the analog transmissions (either with BLUE or averaging algorithm) consume less energy than the digital transmissions. However, when $L \geq 2M$, the digital transmissions consume less energy than

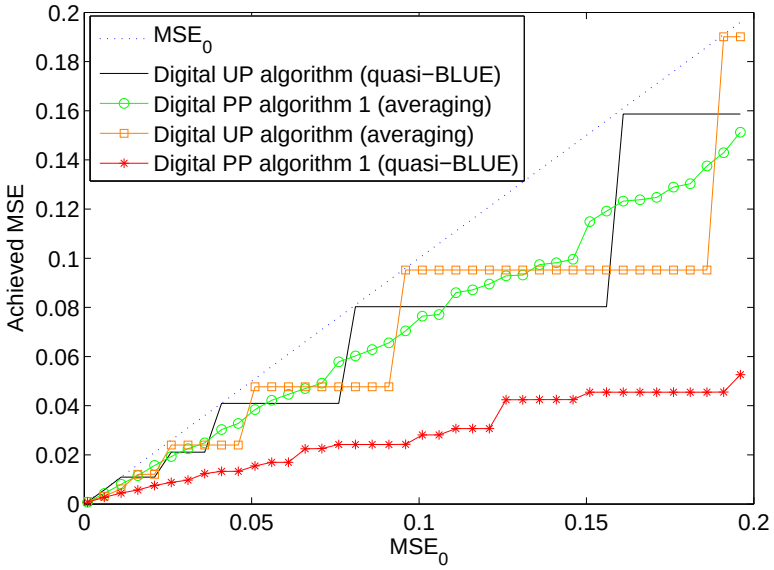

Figure 3.13: For the digital PP and UP algorithms, the actual MSE value at the destination node versus the target MSE MSE_0.

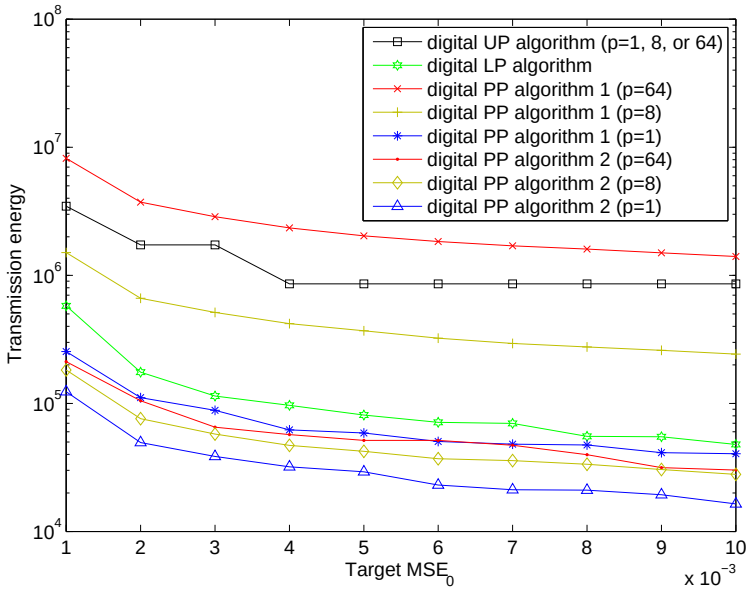

Figure 3.14: The sum energy required by different energy planning algorithms for progressive estimation.

the analog transmissions, which is essentially the same conclusion from Section 3.4 where $M = 1$ was considered. The energy saving by increasing L becomes less significant when L is large, which is consistent with Fig. 3.12.

3.9.5 Progressive versus Single-hop on Energy Cost

In this subsection, we compare the energy cost with progressive and single-hop estimation. By the way, the optimal digital energy planning algorithm for single-hop networks is studied in [13]. Here we consider an example of scalar estimation in this simulation for comparison because the algorithm derived in [13] is only for scalar estimation. We estimate a scalar θ. Local observation model for each sensor k is

$$x_k = \theta + n_k \tag{3.88}$$

where θ is the desired parameter to be estimated. We assume its range is $[-w, w]$ where $w = 1$. n_k is the measurement noise with zero mean and k-dependent variance $\sigma^2_{n_k}$. We assume $\sigma^2_{n_k} = 0.05$ in this simulation. The wireless channel model is assumed same as before except that there is only one sub-channel available between sensors.

Fig. 3.18 shows the total normalized transmission energy consumed by the network versus the target MSE_0. The highest curve is for the optimal digital energy planning algorithm for single-hop networks presented in [13], which is referred as nonprogressive (NP) algorithm. The NP algorithm is conducted over network of the same topology as shown in 3.2. But with NP algorithm, each sensor communicates with the fusion center directly without the assist of the routing tree. The middle curve is for the digital UP algorithm, and the lowest curve is for the digital PP algorithm 1. The NP and UP curves

cross each other at certain values of MSE_0. We see that the PP algorithm 1 requires the least amount of energy throughout the whole MSE_0 region. At high MSE_0, the PP algorithm 1 and the NP algorithm consume approximately the same energy because almost all bits are allocated to the sensors right next to the fusion center.

3.9.6 Robustness

We now consider a situation that after the energy planning is completed, some nodes fail due to some reasons such as running out of battery. But the fusion center is unaware of those failures due to feedback delay. Hence the energy planning will not be updated and each node in the network still uses the originally determined energy to transmit information. We assume that each node fails randomly with the probability p_f. Fig. 3.19 shows the actual MSE at the destination node versus p_f for the digital case, and 3.20 shows the actual MSE at the destination node versus p_f for the analog case. For both cases, the target MSE is 0.002. We see that for both cases, the actual MSE at the destination increases gracefully as p_f increases.

3.10 Summary

In this chapter, we have developed two digital transmission energy planning algorithms and an analog transmission energy planning algorithm for progressive estimation in multihop sensor network with routing tree. The routing tree finding and the transmission energy planning are conducted at the startup of the network or once the network condition changes. The network condition (such as topology and channel state information) is

assumed to be constant during the time of interest for estimating and tracking spatially invariant parameters. These transmission energy planning algorithms guarantee any pre-specified estimation performance at the destination node. And at the same time, they significantly reduce the required transmission energy for the entire network. Unlike many other consensus-type estimation schemes, the proposed progressive estimation schemes along with their energy planning algorithms yield any desired result within a finite time although they require an operational overhead at the startup. The energy planing algorithms shown in this paper provide an optimal energy planning for the proposed progressive estimation algorithms based on averaging. For algorithms based on BLUE or quasi BLUE, the result of our energy planning algorithms show a rather conservative gap. Whether or not this gap can be narrowed by a more clever energy planning algorithm remains a future research topic.

In practice, any sensor in the network can be a destination node, and there could be multiple destination nodes in the network. The theory and technique shown in this paper are applicable to any given destination node and its associated routing tree. We have seen that most of the energy should be distributed relatively near the destination node (unless a large p is used in the L_p norm). This fact should be taken into account when a routing tree is searched for in a large network. Finally, we note that the proposed algorithms are readily applicable to any single-hop network (as a special case) where each sensor transmits data directly to the destination node. In this special case, no routing tree is needed.

3.11 Appendix

3.11.1 Pulse Modulation

Assume a complex baseband channel between two sensors where the input $x(t)$ and the output $y(t)$ are related as follows:

$$y(t) = hx(t) + v(t) \tag{3.89}$$

where h is a complex channel coefficient, $v(t)$ is complex Gaussian noise with the energy spectral density function:

$$S_v(f) = \begin{cases} N_0 & |f| \leq \mathcal{W}/2 \\ 0 & otherwise \end{cases} \tag{3.90}$$

To transmit a complex symbol s over this channel, we can use the pulse amplitude modulation $x(t) = sc(t)$ where the waveform $c(t)$ has the duration $[0, \mathcal{T}]$ and the double-sided bandwidth $\mathcal{W}$. Then, the output of the channel is $y(t) = hsc(t) + v(t)$. To obtain the maximum likelihood estimate of s based on $y(t)$, we can first compute the sufficient statistics $\bar{y} = \int_0^{\mathcal{T}} y(t)c(t)dt$ which is also known as matched filtering. It follows that

$$\bar{y} = hE_c s + \bar{v} \tag{3.91}$$

where E_c is the energy of the waveform $c(t)$ and $\bar{v} = \int_0^{\mathcal{T}} v(t)c(t)dt$ is a complex Gaussian random variable with zero mean and the variance

$$\sigma_v^2 = \mathcal{E}\left\{|\bar{v}|^2\right\} = \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} R_v(\tau - t)c(\tau)c(t)d\tau dt \tag{3.92}$$

where $\mathcal{E}$ denotes expectation, $R_v(\tau)$ is the autocorrelation function of $v(t)$. Since $R_v(\tau) = \int_{-\mathcal{W}/2}^{\mathcal{W}/2} N_0 \exp\{2\pi\tau f\} df$, it follows from (3.92) that

$$\sigma_v^2 = N_0 \int_{-\mathcal{W}/2}^{\mathcal{W}/2} |C(f)|^2 df = N_0 E_c \tag{3.93}$$

where $C(f)$ is the Fourier transform of $c(t)$. Then, the maximum likelihood estimate of s is $\hat{s} = \bar{y}/(hE_c)$ which has the signal to noise ratio:

$$SNR \doteq \frac{|s|^2}{\mathcal{E}\{|\hat{s} - s|^2\}} = \frac{|s|^2|h|^2E_c^2}{\sigma_v^2} = \frac{|s|^2|h|^2E_c}{N_0} = \frac{|h|^2E}{N_0} \tag{3.94}$$

where $E = |s|^2E_c$ is the total transmitted energy, i.e, the energy of $x(t)$ within $[0, \mathcal{T}]$.

3.11.2 Proof of Lemma 1

Proof. We first prove that

$$\left(\sum_{k=1}^{2} \boldsymbol{A}_k^{-1}\right)^{-1} \leq \frac{1}{4}\left(\sum_{k=1}^{2} \boldsymbol{A}_k\right). \tag{3.95}$$

Denote the eigenvalue decomposition $\boldsymbol{A}_1^{H/2}\boldsymbol{A}_2^{-1}\boldsymbol{A}_1^{1/2} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$. Then, (3.95) is equivalent to each of the following inequalities:

$$\boldsymbol{A}_1^{-1/2}(\boldsymbol{A}_1^{-1} + \boldsymbol{A}_2^{-1})^{-1}\boldsymbol{A}_1^{-H/2} \leq \boldsymbol{A}_1^{-1/2}\left(\frac{\boldsymbol{A}_1 + \boldsymbol{A}_2}{4}\right)\boldsymbol{A}_1^{-H/2} \tag{3.96}$$

$$\left(I + \boldsymbol{A}_1^{H/2}\boldsymbol{A}_2^{-1}\boldsymbol{A}_1^{1/2}\right)^{-1} \leq \frac{1}{4}\left(I + \boldsymbol{A}_1^{-1/2}\boldsymbol{A}_2\boldsymbol{A}_1^{-H/2}\right) \tag{3.97}$$

$$\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{\Lambda})^{-1}\boldsymbol{U}^H \leq \frac{1}{4}\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{\Lambda}^{-1})\boldsymbol{U}^H \tag{3.98}$$

$$(\boldsymbol{I} + \boldsymbol{\Lambda})^{-1} \leq \frac{1}{4}(\boldsymbol{I} + \boldsymbol{\Lambda^{-1}}) \tag{3.99}$$

$$\frac{1}{1 + \lambda_i} \leq \frac{1}{4}(1 + \frac{1}{\lambda_i}), \quad \forall i \tag{3.100}$$

where $\lambda_i > 0$ is the ith element of the diagonal matrix $\boldsymbol{\Lambda}$. It is easy to verify that (3.100) is equivalent to $(\lambda_i - 1)^2 \geq 0$ which holds always.

We now assume that Lemma 1 holds for $n = m \geq 2$, i.e.,

$$\left(\sum_{k=1}^{m} \boldsymbol{A}_k\right)\left(\sum_{k=1}^{m} \boldsymbol{A}_k^{-1}\right) \geq m^2 \boldsymbol{I} \tag{3.101}$$

Then, we can write

$$\begin{aligned}
&\left(\sum_{k=1}^{m} \boldsymbol{A}_k + \boldsymbol{A}_{m+1}\right)\left(\sum_{k=1}^{m} \boldsymbol{A}_k^{-1} + \boldsymbol{A}_{m+1}^{-1}\right) \\
&= \left(\sum_{k=1}^{m} \boldsymbol{A}_k\right)\left(\sum_{k=1}^{m} \boldsymbol{A}_k^{-1}\right) + \left(\sum_{k=1}^{m} \boldsymbol{A}_k\right)\boldsymbol{A}_{m+1}^{-1} + \boldsymbol{A}_{m+1}\left(\sum_{k=1}^{m} \boldsymbol{A}_k^{-1}\right) + I \\
&\geq m^2 \boldsymbol{I} + \boldsymbol{C} + \boldsymbol{D} + \boldsymbol{I}
\end{aligned} \tag{3.102}$$

where $\boldsymbol{C} = \left(\sum_{k=1}^{m} \boldsymbol{A}_k\right)\boldsymbol{A}_{m+1}^{-1}$ and $\boldsymbol{D} = \boldsymbol{A}_{m+1}\left(\sum_{k=1}^{m} \boldsymbol{A}_k^{-1}\right)$. Then, it follows that

$$\begin{aligned}
\boldsymbol{C} + \boldsymbol{D} + 2m\boldsymbol{I} &= \left(\sum_{k=1}^{m}\left(\boldsymbol{A}_k\boldsymbol{A}_{m+1}^{-1} + \boldsymbol{A}_{m+1}\boldsymbol{A}_k^{-1}\right)\right) + 2m\boldsymbol{I} \\
&= \sum_{k=1}^{m}\left(\boldsymbol{A}_k\boldsymbol{A}_{m+1}^{-1} + \boldsymbol{A}_{m+1}\boldsymbol{A}_k^{-1} + 2I\right) \\
&= \sum_{k=1}^{m}\left(\boldsymbol{A}_k\boldsymbol{A}_{m+1}^{-1} + \boldsymbol{A}_{m+1}\boldsymbol{A}_k^{-1} + \boldsymbol{A}_k\boldsymbol{A}_k^{-1} + \boldsymbol{A}_{m+1}\boldsymbol{A}_{m+1}^{-1}\right) \\
&= \sum_{k=1}^{m}(\boldsymbol{A}_k + \boldsymbol{A}_{m+1})(\boldsymbol{A}_k^{-1} + \boldsymbol{A}_{m+1}^{-1}) \\
&\geq 4m\boldsymbol{I}
\end{aligned} \tag{3.103}$$

where the last inequality is due to (3.95). Using (3.103) in (3.102) yields

$$\left(\sum_{k=1}^{m} \boldsymbol{A}_k + \boldsymbol{A}_{m+1}\right)\left(\sum_{k=1}^{m} \boldsymbol{A}_k^{-1} + \boldsymbol{A}_{m+1}^{-1}\right) \geq (m+1)^2 \boldsymbol{I} \tag{3.104}$$

By the above induction, Lemma 1 is proven. ■

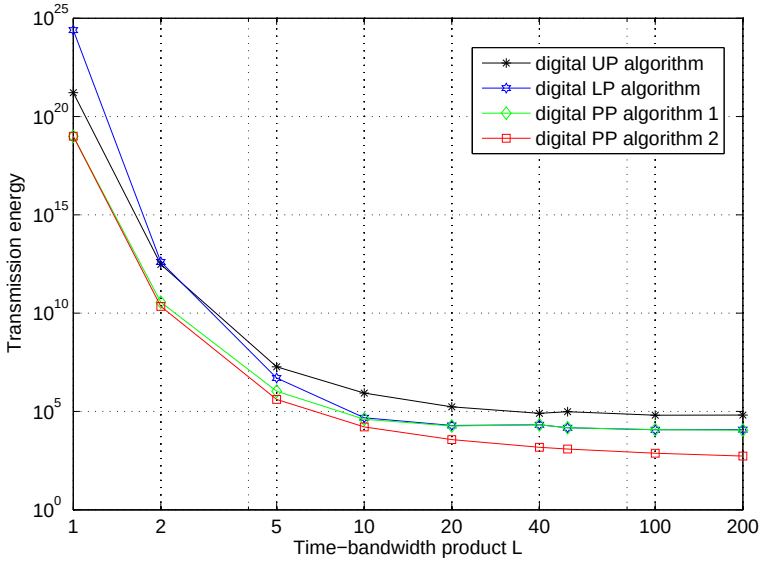

Figure 3.15: The sum energy required by four different energy planning algorithms for progressive estimation versus the time-bandwidth product L, where $MSE_0 = 0.01$. For digital PP algorithm 1 and 2, $p = 1$ was used.

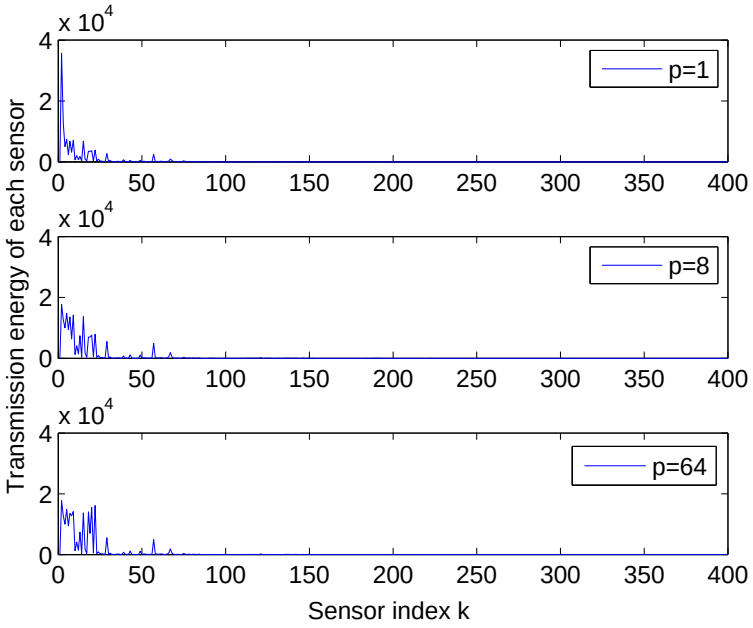

Figure 3.16: Amount of transmission energy consumed by an individual sensor versus the sensor index for the digital PP algorithm 2 for progressive estimation, with $p = 1$, 8 and 64, where $MSE_0 = 0.001$. The sensor index is sorted increasingly as the distance between the sensor and the fusion center increases. For progressive estimation, sensor zero in this and the following figures is the fusion center.

.

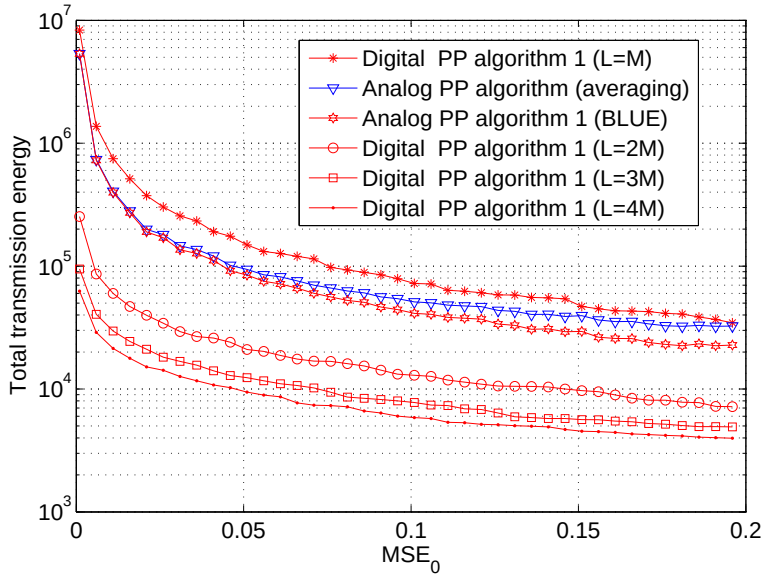

Figure 3.17: Comparison between analog transmissions and digital transmissions: total transmission energy versus the target MSE.

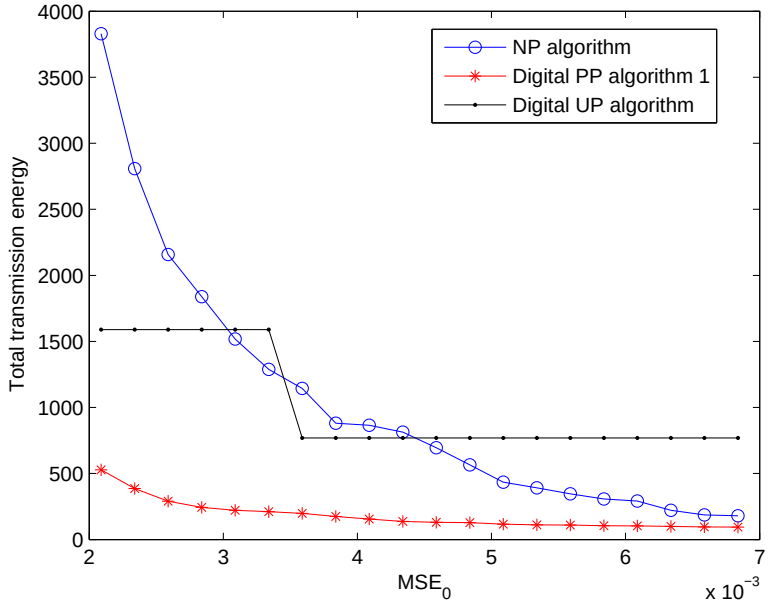

Figure 3.18: Total amount of normalized transmission energy consumed by the network versus MSE_0. The network used is Fig. 3.2.

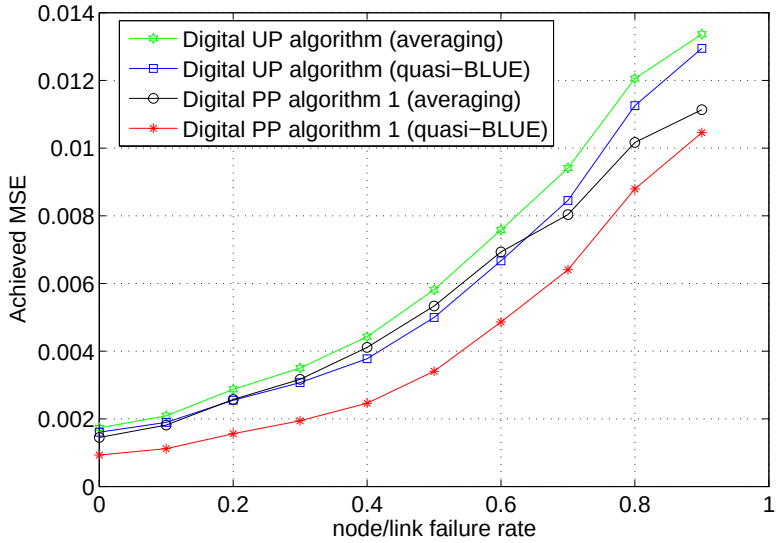

Figure 3.19: Digital transmissions: Actual MSE versus node failure rate. The target MSE is $MSE_0 = 0.002$.

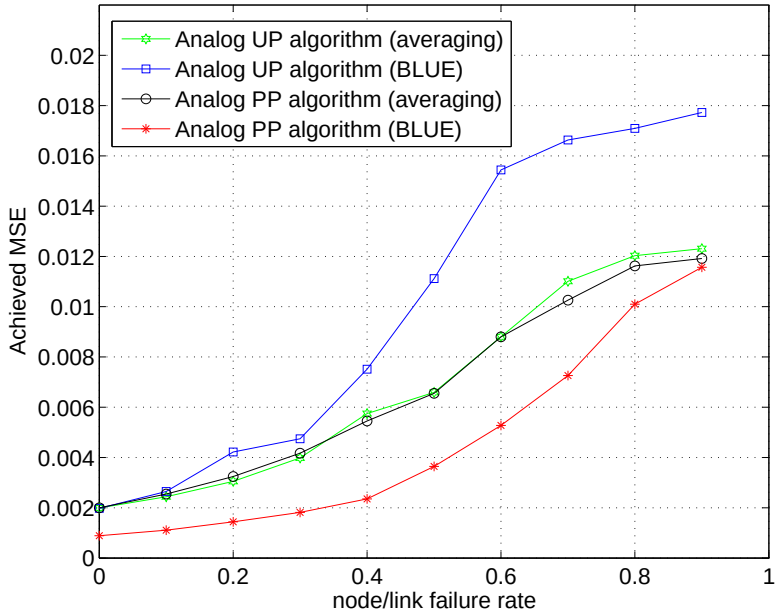

Figure 3.20: Analog transmissions: Actual MSE versus node failure rate. The target MSE is $MSE_0 = 0.002$.

Chapter 4

Energy Planning for Consensus Data Fusion

4.1 System Model

In this chapter, we consider the same sensor network as in previous chapter. The desired unknown deterministic vector is denoted by $\boldsymbol{\theta}^{(n)} \in R^{M\times 1}$, where n denotes the sampling time. As in previous chapter, we assume that all data observation, computations, and communications required for estimating $\boldsymbol{\theta}^{(n)}$ are performed within the time window from the sampling time n to the sampling time $n+1$. Because of this reason, we will drop the index n in $\boldsymbol{\theta}^{(n)}$ for convenience.

The data collected by each sensor $\boldsymbol{x}_k$ can be of any dimension and any statistical properties. But we assume that each sensor (sensor k) is able to use its collected data to obtain an original estimate $\boldsymbol{\theta}_k(0) \in R^{M\times 1}$. We assume that $\boldsymbol{\theta}_k(0)$ is unbiased and has a

bounded covariance matrix, i.e., $E\{\boldsymbol{\theta}_k(0)\} = \boldsymbol{\theta}^{(n)}$ and $E\{\boldsymbol{\theta}_k(0)\boldsymbol{\theta}_k(0)^T\} \leq \boldsymbol{C}_{\theta,k}$.

Consensus estimation essentially is a iterative in-network computation with information sharing among neighboring sensors. At iteration time t, we denote the estimate at the kth sensor of the unknown vector $\boldsymbol{\theta}$ as $\boldsymbol{\theta}_k(t)$. Denote the set of neighbors of sensor k as $\mathcal{N}_k$. $\boldsymbol{\theta}_k(t)$ can be calculated at sensor k by using the estimates $\boldsymbol{\theta}_i(t-1)$ where i belongs to the neighbor set of k, i.e., $i \in \mathcal{N}_k$.

$$\boldsymbol{\theta}_k(t) = f(\boldsymbol{\theta}_k(t-1), \cdots, \boldsymbol{\theta}_i(t-1), \cdots), i \in \mathcal{N}_k \tag{4.1}$$

Here f is the fusion algorithm need to be designed, which fuses the multiple estimates in a neighborhood to generate a better estimation of $\boldsymbol{\theta}$. The principle of the consensus estimation is quite simple. In every iteration t, each node k first gathers estimates $\boldsymbol{\theta}_i(t-1)$ from its neighborhood by wireless communication technique which will be discussed in details later, and then runs the fusion algorithm to improve its local estimation. It has been proved that as long as the network is fully connected and some other conditions are met, the local estimates $\boldsymbol{\theta}_k(t)$ will converge to a consensus when t is sufficiently large. (See [] and more references therein for the conditions of convergence and characteristics of convergence.) In other words, all sensors in the network will give same estimation result eventually, i.e., $\boldsymbol{\theta}_k(\infty) = \boldsymbol{\theta}^*$. We treat $\boldsymbol{\theta}^*$ as our final estimation for $\boldsymbol{\theta}$ with the consensus estimation framework.

For the consensus estimation, it is important to note that the iteration time t is different from the sampling time n. Typically, from the sampling time n to the sampling time $n+1$, many iterations are required by a consensus algorithm. Like in the previous chapter for progressive estimation, we consider computations within a single window between two

adjacent sampling instants.

Essentially, consensus estimation is a combination problem of communication and signal processing. The communication part includes the exchange of information between neighboring sensors and the signal processing includes applying fusion algorithms to process information locally at each individual sensor. We study the two aspects in details next.

4.2 Communication Protocol

Designing distributed communication protocols for consensus estimation over a set of sensors by allowing information sharing between neighboring sensors has been studied intensively recently [35], [36], [37], [38], [39]. A very good review of these protocols can be found in [40]. The basic principle of those protocols in [35], [36], [37], [38], [39] is to apply asynchronously distributed scheduling by randomize nodes' transmission time to allow nodes to exchange information with their one-hop neighbors with less or no collisions. In this section, following this principle, a broadcasting type of protocol is applied to let sensors communicate with each other. A similar broadcast gossip algorithms for consensus estimation was proposed in [39]. However, comparing to the gossip algorithm and the other protocols in the literature, this broadcasting protocol design includes more fundamental and practical issues such as transmission energy, number of quantization bits, and condition of success packet reception, which have been overlooked before.

For each iteration t, we denote $\boldsymbol{\theta}_k(t)$ to be the estimation of $\boldsymbol{\theta}$ at sensor k. Assuming that the mth entry of $\boldsymbol{\theta}_k(t)$ is quantized into B_m bits using the probabilistic uniform quantization as described in previous chapter, the quantized version of $\boldsymbol{\theta}_k(t)$ can be written

as

$$\boldsymbol{m}_k(t) = \boldsymbol{\theta}_k(t) + \boldsymbol{n}_k(t), \tag{4.2}$$

where $\boldsymbol{n}_k(t)$ is the quantization error vector with zero mean and a diagonal covariance matrix $\boldsymbol{C}_{q,k}$. As discussed before, the m-th diagonal element of $\boldsymbol{C}_{q,k}$ is upper bounded by $\frac{4W_m^2}{2^{2B_m}}$.

After the above quantization, sensor k broadcasts $\boldsymbol{m}_k(t)$ to its neighboring sensors. We denote set $\mathcal{N}_k^r$ as the set containing all neighboring sensors received $\boldsymbol{m}_k(t)$ successfully. One should notice that, in this consensus estimation application, broadcasting is more energy efficient to sharing information among neighbors than peer-to-peer transmission, because only one transmission is enough to send the $\boldsymbol{m}_k(t)$ to all sensors in $\mathcal{N}_k^r$, while $|\mathcal{N}_k^r|$ transmissions are needed if peer-to-peer transmission is applied. Of course, in practice, there can be an additional amount of energy required to avoid collisions between broadcasted packets from different sensors. This additional energy can be kept small especially if an additional bandwidth is available for sensors to self-coordinate their broadcasts. We will ignore this overhead.

As we discussed above, at each iteration t, each sensor k broadcasts a data packet containing $\sum_{m=1}^{M} B_m$ bits. Assuming L sub-channels for each pair of transmitting and receiving sensors, sensor j can successfully receive the data packets broadcasted from sensor k if the following holds:

$$\sum_{l=1}^{L} \psi log_2 \left(1 + \phi\mu \frac{E_k |h_{j,k,l}|^2}{LN_0}\right) \geq \sum_{m=1}^{M} B_m \tag{4.3}$$

where $h_{j,k,l}$ is the fading factor of the sub-channel l from sensor k to sensor j. E_k is the

transmission energy for broadcasting at sensor k. The other parameters are the same as those described for (3.35).

One should notice that E_k and B_m are the two key parameters to govern the neighborhood range of sensor k, i.e., $\mathcal{N}_k^r$. Generally speaking, increasing E_k will increase the neighborhood range of k and increase B_m will decrease the range. The neighborhood range of every sensor actually determines the network connectivity or in other words the graph of the network. We will show later that by adjusting E_k and B_m, we actually consider consensus estimation over optimized network graph, which is a contribution of this thesis beyond the papers in the literature [26]-[45].

If the sensor distributions are statistically uniform, it is reasonable for each sensor to use the same amount of energy E_0 for each broadcast. If the sensor distributions are highly non-unform, it may be a good idea for each sensor to use a different amount of energy in order to reach a certain number of neighbors. Clearly, depending on the sensor distributions, different sensors may need to consume different amounts of energy. But for convenience, we will assume that each sensor uses the same amount of energy E_0 for each broadcast.

The total energy consumed by the network can be written as $E_{total} = TKE_0$ where K is the total number of sensors in the network and T is the number of iterations required until the required performance is achieved. If we increase E_k, then the size of the neighborhood of each sensor increases and hence T is expected to decrease. One extreme case is to set E_0 very high such that all sensors can hear from each other. Hence only one round of broadcasting from each sensor is enough to achieve consensus among sensors,

i.e. $T = 1$. Likewise, if we decrease E_0, T is expected to increase due to the shrinking of neighborhood in each iteration. To minimize E_{total}, there is an optimal choice of E_0.

4.3 Collision Avoidance

To avoid the collision and strong co-channel interference between the broadcasting for different sensors, we propose a distributed scheduling method called distributed synchronized array method (D-SAM).

In D-SAM, time is slotted into frames of equal duration as shown in Fig. 4.1. Each frame is further divided into control subframe and data subframe. Assuming that the data subframe is much longer than the control subframe, the network spectral efficiency is dominated by the spectral efficiency in each data subframe. A control subframe is used for each node to compete for data transmission opportunity in a data subframe. Each control subframe consists of a group of M contention slots.

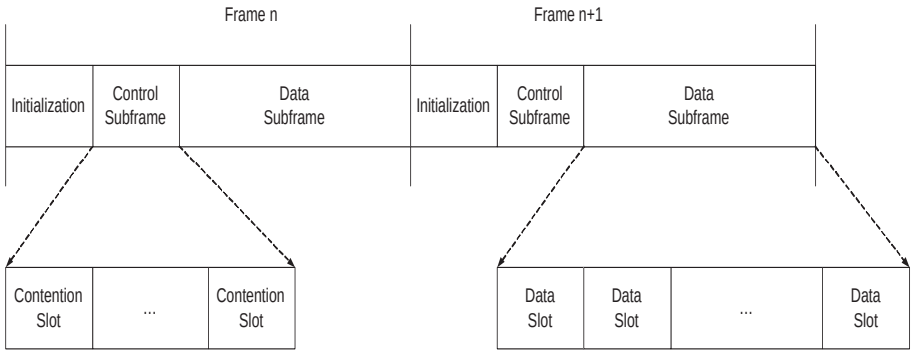

Figure 4.1: Frame structure of the distributed SAM protocol, which resembles that of MSH-DSCH in IEEE 802.16.

At the beginning of each frame, D-SAM allows each node to randomly initialize

a choice for one of the M contention slots if the node has a packet to transmit to another node. If the node has packets to be transmitted (separately) to multiple neighboring nodes, the node chooses multiple contention slots - one slot for each receiver. During a chosen contention slot, the node contends for the upcoming data subframe by starting a handshaking process with its intended receiving node. The handshaking involves three packets: RTS (request-to-sent), CTS (clear-to-sent), and ACK. If the handshaking is successful, the upcoming data subframe is reserved for data transmission between the transmitter-and-receiver pair. During each contention slot, the handshaking packets are received by neighboring nodes so that these nodes are aware of the reservation status of the upcoming data frame. For each frame and each neighborhood in a predetermined range, the data subframe can only be reserved for one transmitter-and-receiver pair. This means that the first contention slot has the highest priority, the second contention slot has the second priority, and so on. In the next frame, the contention process repeats without memory of the previous contentions, which ensures fairness.

More details of D-SAM are as follows. We assume that each node k maintains a neighborhood list $\mathscr{N}_k$ which contains the identifications of all its neighboring nodes inside a cooperative range $\mathscr{R}$. The range $\mathscr{R}$ is an important parameter for the performance of D-SAM. The ith node in $\mathscr{N}_k$ is indexed by $\mathscr{N}_k(i)$. The neighborhood list at each node can be established at the startup of the network. For networks of low mobility, this startup is feasible. We assume that every node can be set to one of three states for the upcoming data subframe: T, R and S. Here, T stands for transmitting, R for receiving, and S for standby. We denote the state of node k as $\mathscr{S}_k$ and the state of $\mathscr{N}_k(i)$ as $\mathscr{S}_{\mathscr{N}_k(i)}$.

1. *Initialization*: At the beginning of each frame, set every node to state S, i.e., $\mathscr{S}_k = S$ for all k. Then, we allow that every node k generates a "contention request vector" $\boldsymbol{v}_k$ that randomly maps each neighboring node in list $\mathscr{N}_k$ to one of M contention slots if node k has a broadcast packet intended to those neighbors. Here, we assume M is larger than the size of every neighbor list, i.e., $M \geq |\mathscr{N}_k|$ for all k. The ratio of M over $|\mathscr{N}_k|$ affects the probability of handshaking collisions. The larger is the ratio, the lower is the probability of handshaking collisions. We denote the mth element of $\boldsymbol{v}_k$ as $v_k(m)$, which is

$$v_k(m) = \begin{cases} j, & \text{if node } k \text{ has traffic to node } j, \text{ and } j \text{ is mapped into contention slot } m, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

   In other words, the value of $v_k(m)$ is the index of the receiving node for which the transmitting node k wants to contend during the contention slot m for the upcoming data subframe. If $v_k(m) = 0$, it means that, in the mth contention slot, node k will not contend for the upcoming data subframe.

2. *In contention slot* m: each node k will first check its contention request vector. If $v_k(m) = 0$, node k eavesdrops ongoing handshaking within the neighborhood of range $\mathscr{R}$. (Naturally, we assume that the eavesdropping range $\mathscr{R}_e$ from each node is larger than the cooperative range $\mathscr{R}$. Furthermore, the carrier sense range $\mathscr{R}_c$, although not considered in this thesis, would be even larger than $\mathscr{R}_e$.) If node k hears any CTS or ACK packet, it retrieves the information from the packet and resets the states of the nodes in $\mathscr{N}_k$ accordingly. If $v_k(m) = j$ where $j > 0$, node k will try to finish the

following three-way RTS-CTS-ACK handshaking with node j:

- RTS

  Node k sends a RTS packet to node j which contains the identity of node k, if the following conditions are satisfied:

  (a) $\mathscr{S}_k = S$.

  (b) $\mathscr{S}_{\mathscr{N}_k(i)} \neq R$ for all i.

- CTS

  If node j has successfully received the RTS packet from node k and the following conditions are satisfied:

  (a) $\mathscr{S}_j = S$

  (b) $\mathscr{S}_{\mathscr{N}_j(i)} = S$ for all i

  (c) $v_j(m) = 0$

  then node j resets $\mathscr{S}_j = R$ and $\mathscr{S}_{\mathscr{N}_j(i_k)} = T$ where i_k is the index for node k in the table $\mathscr{N}_j$, and sends a CTS packet back to node k.

- ACK

  If node k has successfully received the CTS packet from node j, the node k resets $\mathscr{S}_k = T$ and $\mathscr{S}_{\mathscr{N}_k(i_j)} = R$ where i_j is the index of node j in the table $\mathscr{N}_k$, and sends back the ACK packet.

During any contention slot, if there is a collision of control packets within the radius $\mathscr{R}$, the operation in that slot is abandoned. But the collision of control packets at

a receiver due to transmitters outside the radius $\mathscr{R}$ is assumed to be resolvable by using coding with relatively high redundancy. If the ratio of M over the number of nodes within the radius $\mathscr{R}$ is large, the probability of collision of control packets is small. As long as the control packets are much smaller than the data packets (i.e., the control subframe is much smaller than the data subframe), the energy efficiency is still high.

Fig. 4.2 illustrates a snapshot of the concurrent co-channel transmission pairs for a square network, which was determined by D-SAM for data transmission. The radius $\mathscr{R} = d_a$ was chosen, where d_a is the spacing between two nearest neighbors. The number of contention slots was $M = 8$. To decrease the co-channel interference, once can just simply increase $\mathscr{R}$.

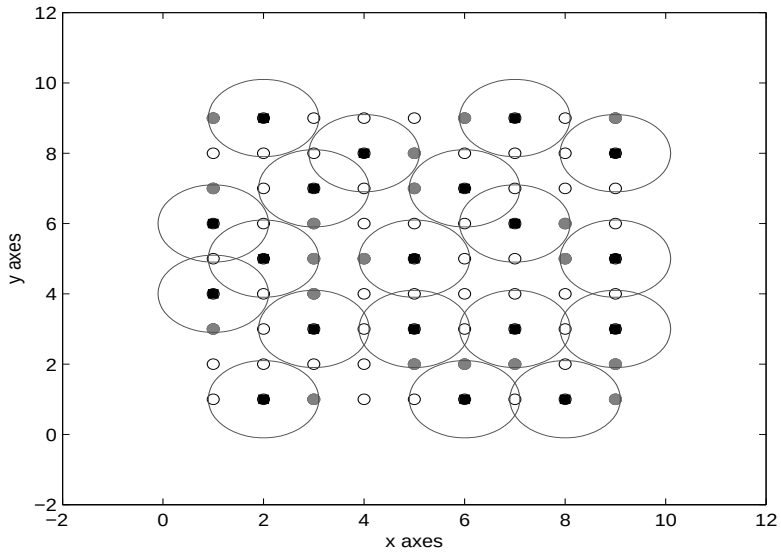

Figure 4.2: A snapshot of concurrent co-channel transmissions determined by the D-SAM protocol for a network in a regular square grid. $\mathscr{R} = d_a$, $M = 8$, where d_a is the minimum distance between two adjacent nodes. The black nodes are the receiving nodes, and the grey nodes are the transmitting nodes.

4.4 Fusion Algorithms

Fusion algorithm is the algorithm each sensor utilizes to update its local estimate of $\boldsymbol{\theta}$ by fusing the received information from neighbor sensors. A general fusion algorithm is presented recently and the conditions for convergence to consensus is given in [40]. In the paper, we apply fusion algorithms which perform constant weighting linear combination of previous local estimate and received estimates from neighboring sensors. More importantly, we study their energy planning performance instead of their convergence conditions and rates.

The fusion algorithm (quantize-after-fusion) that we will first consider is:

$$\boldsymbol{\theta}_k(t+1) = W_{k,k}\boldsymbol{\theta}_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}\boldsymbol{m}_j(t) \tag{4.5}$$

where $\mathcal{N}_k$ is the set of the indices of the neighboring transmitting sensors from which sensor k successfully receives a quantized vector, $\boldsymbol{m}_j(t)$ is a quantized version of $\boldsymbol{\theta}_j(t)$ from sensor j, and $W_{k,j}$ are scalar weights.

We can write (4.5) for all k in a vector form as

$$\boldsymbol{\theta}(t+1) = (\boldsymbol{W}\otimes\boldsymbol{I})\boldsymbol{\theta}(t) + (\boldsymbol{V}\otimes\boldsymbol{I})\boldsymbol{n}(t) \tag{4.6}$$

where $\boldsymbol{I}$ is the $M\times M$ identity matrix, $\otimes$ is the Kronecker product, $(\boldsymbol{W})_{k,j} = W_{k,j}$, $\boldsymbol{V} = \boldsymbol{W} - \boldsymbol{W}_D$, $\boldsymbol{W}_D$ is a diagonal matrix with the same diagonal elements as $\boldsymbol{W}$, $\boldsymbol{\theta}(t) = (\boldsymbol{\theta}_1(t)^T,\cdots,\boldsymbol{\theta}_K(t)^T)^T$, and $\boldsymbol{n}(t) = (\boldsymbol{n}_1(t)^T,\cdots,\boldsymbol{n}_K(t)^T)^T$. One should notice that the above fusion algorithm is different from the one applied in [42] where the perturbation term at each fusion iteration is independent of the weighting matrix $\boldsymbol{W}$. While the perturbation term in algorithm 1 is $(\boldsymbol{V}\otimes\boldsymbol{I})\boldsymbol{n}(t)$, which depends on $\boldsymbol{W}$.

In the absence of the quantization noise, we want $\lim_{t\to\infty}\boldsymbol{\theta}_i(t) = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\theta}_k(0)$ for all k. As shown in [28], this is equivalent to $\lim_{t\to\infty}\boldsymbol{W}^t = \frac{\mathbf{1}\mathbf{1}^T}{K}$ where $\mathbf{1}$ is a $K\times 1$ vector consisting of all 1s. This is also equivalent to the following conditions: $\mathbf{1}^T\boldsymbol{W} = \mathbf{1}^T$, $\boldsymbol{W}\mathbf{1} = \mathbf{1}$, and $\rho(\boldsymbol{W} - \frac{\mathbf{1}\mathbf{1}^T}{K}) < 1$ where $\rho(\boldsymbol{A})$ denotes the spectral radius of the matrix $\boldsymbol{A}$. We will assume that $\boldsymbol{W}$ always meets the above conditions.

With the above conditions, one can verify that the mean of $\boldsymbol{\theta}(t)$ is given by $\mathcal{E}\{\boldsymbol{\theta}(t)\} = \mathbf{1}\otimes\boldsymbol{\theta}$, and the MSE of $\boldsymbol{\theta}(t)$ is given by

$$\begin{aligned} MSE(t) &\doteq Tr\left(\mathcal{E}\left\{(\boldsymbol{\theta}(t)-\mathbf{1}\otimes\boldsymbol{\theta})(\boldsymbol{\theta}(t)-\mathbf{1}\otimes\boldsymbol{\theta})^T\right\}\right) \\ &\le Tr\left\{\boldsymbol{D}_c(\boldsymbol{W}^t)^T\boldsymbol{W}^t\right\} + \left(\sum_{i=0}^{t-1} Tr\left\{(\boldsymbol{W}^i\boldsymbol{V})(\boldsymbol{W}^i\boldsymbol{V})^T\right\}\right)\left(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}}\right) \doteq MSE_U(t) \end{aligned} \tag{4.7}$$

where $\boldsymbol{D}_c$ is a diagonal matrix whose (k,k)-th entry is $Tr\{\boldsymbol{C}_{\theta,k}\}$. Here, we have used $E\{(\boldsymbol{\theta}_k(0)-\boldsymbol{\theta})(\boldsymbol{\theta}_k(0)-\boldsymbol{\theta})^T\} = \boldsymbol{C}_{\theta,k}$, $(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D}) = \boldsymbol{A}\boldsymbol{C}\otimes\boldsymbol{B}\boldsymbol{D}$, $(\boldsymbol{A}\otimes\boldsymbol{B})^T = \boldsymbol{A}^T\otimes\boldsymbol{B}^T$, and $Tr\{\boldsymbol{A}\otimes\boldsymbol{B}\} = Tr\{\boldsymbol{A}\}Tr\{\boldsymbol{B}\}$. $MSE_U(t)$ is an upper bound on $MSE(t)$. Due to the fact that quantization error is driven to zero as the consensus is achieved [41], this upper bound becomes loose when t grows.

Besides the quantize-after-fusion algorithm, there are other fusion algorithms in the literature. In [43], a "quantize-before-subtract" algorithm is presented and its vector extension form can be written into

$$\boldsymbol{\theta}_k(t+1) = \boldsymbol{\theta}_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}\left(\boldsymbol{m}_j(t) - \boldsymbol{m}_k(t)\right) \tag{4.8}$$

In [41], a "quantize-before-fusion" algorithm is considered in the following form

$$\boldsymbol{\theta}_k(t+1) = W_{k,k}\boldsymbol{m}_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}\boldsymbol{m}_j(t) \tag{4.9}$$

After uniform probabilistic quantization, the local estimation is updated to

$$\boldsymbol{m}_k(t+1) = W_{k,k}\boldsymbol{m}_k(t) + \sum_{j\in\mathcal{N}_k} W_{k,j}\boldsymbol{m}_j(t) + \boldsymbol{n}_k(t+1) \tag{4.10}$$

where $\boldsymbol{n}_k(t+1)$ is the quantization noise for iteration $n+1$.

It has been proved in [41] that the quantize-before-fusion can converge to a consensus at one of the quantization levels. With the quantize-after-fusion or the quantize-before-subtract algorithms, one can easily check the quantization error at each iteration is smaller than with the quantize-before-fusion algorithm. Therefore, when applying the same probabilistic quantization, these two algorithms should also converge to consensus. Furthermore, it is straightforward to see the quantization error with the quantize-after-fusion algorithm is smaller than with the quantize-before-subtract algorithm. Therefore, we will utilize the quantize-after-fusion algorithm as the fusion algorithm in the rest of this paper.

4.5 Energy Planning

4.5.1 Problem Formulation and Simplification

Parallel to the energy planning for progressive estimation, we study the energy planning algorithm for consensus estimation in this section. The purpose of energy planning to optimize E_0, B_m, and $\boldsymbol{W}$ to minimize the broadcasting energy consumed by all sensors until the mean square error of the estimation is smaller than a threshold MSE_0.

Mathematically the energy planning problem for consensus estimation can be formulated as follows

$$\min_{t,E_0,B_m,\boldsymbol{W}} : tKE_0 \tag{4.11}$$

subject to:

$$MSE_j(t, \boldsymbol{W}, E_0, B_m)/K \leq MSE_0 \tag{4.12}$$

$$\mathbf{1}^T \boldsymbol{W} = \mathbf{1}^T \tag{4.13}$$

$$\boldsymbol{W}\mathbf{1} = \mathbf{1} \tag{4.14}$$

$$\rho(\boldsymbol{W} - \frac{\mathbf{1}\mathbf{1}^T}{K}) < 1 \tag{4.15}$$

$$t \geq 0 \tag{4.16}$$

where $j = 1, 2,$ or 3, which corresponding to different fusion algorithms presented in previous section.

The above optimization problem in general is a very difficult one. We then attempt to simplify the problem by choosing special B_m and $\boldsymbol{W}$. We assume that $B_m = B$ for all m. We also select $\boldsymbol{W}$ following the principle of constant edge weights [28], i.e., $\boldsymbol{W} = \boldsymbol{I} - \gamma \boldsymbol{L}$, where $\boldsymbol{L}$ is the Laplacian matrix of the network graph as defined below

$$(\boldsymbol{L})_{j,k} = \begin{cases} -1, & j \in \mathcal{N}_k \\ |\mathcal{N}_k|, & k = j \\ 0, & otherwise \end{cases} \tag{4.17}$$

With this selection of $\boldsymbol{W}$, it is easy to see (4.13) and (4.14) are automatically satisfied. Condition (4.15) is equivalent to $0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})}$ as shown in [28]. Here $\lambda_1(\boldsymbol{L})$ is the largest eigenvalue of matrix $\boldsymbol{L}$.

With the above simplifications, the total cost of energy can be minimized over the

variables E_0, B and γ. In other words, we need to solve the following problem:

$$\min_{t,E_0,B,\gamma} : tKE_0 \tag{4.18}$$

$$\text{s.t.} \quad MSE_j(t,\gamma,E_0,B)/K \leq MSE_0 \tag{4.19}$$

$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})} \tag{4.20}$$

$$t \geq 0 \tag{4.21}$$

where MSE_0 is the target MSE per sensor, and the solution of t here is the number of iterations required. Note that $MSE(t, E_0, B, \gamma)$ also explicitly depends on E_0 and B, which determines the network Laplacian Matrix $\boldsymbol{L}$.

We propose a two-loop search algorithm to this problem. In the inner loop, we fix E_0 and B, and look for the optimal γ and the associated t, which is to optimize the local fusion algorithm at each sensor. In the outer loop, we search for the optimal E_0 and B, which is to optimize the network connectivity or the Laplacian Matrix $\boldsymbol{L}$. For each outer loop, there are only two variables, and the brute force search for E_0 and B is feasible. However, for the inner loop, a refined algorithm is shown next.

4.5.2 Optimal Selection of γ

With a given pair of E_0 and B, $\mathcal{N}_k$ for all k are determined, and hence so is $\boldsymbol{L}$. The problem of the inner loop search can be formulated as:

$$\min_{\gamma,t} : t \tag{4.22}$$

$$\text{s.t.} \quad MSE_U(t,\gamma)/K \leq MSE_0 \tag{4.23}$$

$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})} \tag{4.24}$$

$$t \geq 0 \tag{4.25}$$

For the quantize-after-fusion algorithm, we can rewrite (4.7) as

$$MSE_U(t,\gamma) = Tr\left\{\boldsymbol{D}_c\boldsymbol{U}\boldsymbol{\Lambda}^{2t}\boldsymbol{U}^T\right\} + \gamma^2 Tr\left\{\boldsymbol{U}\left(\sum_{i=0}^{t-1}\boldsymbol{\Lambda}^{2i}\right)\boldsymbol{U}^T(\boldsymbol{L}-\boldsymbol{L}_D)^2\right\}\left(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}}\right) \tag{4.26}$$

where we have used the eigenvalue decomposition $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, $(\boldsymbol{\Lambda})_{l,l} = \lambda_l(\boldsymbol{W})$, and $\boldsymbol{L}_D$ is the diagonal part of $\boldsymbol{L}$. Also recall that $\lambda_1(\boldsymbol{W}) = 1$ and $|\lambda_l(\boldsymbol{W})| < 1$ for $l \geq 2$. We can then write

$$MSE_U(t,\gamma) = Tr\left\{\boldsymbol{D}_c\boldsymbol{U}\boldsymbol{\Lambda}^{2t}\boldsymbol{U}^T\right\} + \gamma^2 Tr\left\{\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T(\boldsymbol{L}-\boldsymbol{L}_D)^2\right\}\left(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}}\right) \tag{4.27}$$

where $\boldsymbol{D}$ is diagonal, $(\boldsymbol{D})_{1,1} = t$, and $(\boldsymbol{D})_{l,l} = \frac{1-\lambda_l^{2t}(\boldsymbol{W})}{1-\lambda_l^2(\boldsymbol{W})}$ for $l \geq 2$.

With the above expressions, $MSE_U(t,\gamma)$ can be treated as a continuous function of t to simplify the problem. It is easy to check that $MSE_U(t)$ has two components: $\mathcal{O}(t) = Tr\left\{\boldsymbol{D}_c\boldsymbol{U}\boldsymbol{\Lambda}^{2t}\boldsymbol{U}^T\right\}$ is a decreasing function of t, and the second term $\mathcal{Q}(t) = \gamma^2 Tr\left\{\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T(\boldsymbol{L}-\boldsymbol{L}_D)^2\right\}\left(\sum_{m=1}^{M}\frac{4W_m^2}{2^{2B_m}}\right)$ is an increasing function of t. The behavior of $MSE_U(t,\gamma)$ with respect to γ is not as clear. But knowing $\boldsymbol{W} = \boldsymbol{I} - \gamma\boldsymbol{L}$, we know that $\boldsymbol{\Lambda}$ is

a linear function of γ and hence according to (4.27), $MSE_U(t,\gamma)$ is a polynomial function of γ of degree $2t$.

The Lagrangian function of this problem is written down as

$$\mathcal{L} = t + \mu(MSE_U(t,\gamma) - MSE_0 K) + \nu_1(-\gamma) + \nu_2(\gamma - \frac{2}{\lambda_1(\boldsymbol{L})}) + \nu_3(-t) \tag{4.28}$$

and the KKT conditions are

$$1 + \mu\frac{\partial MSE_U(t,\gamma)}{\partial t} - \nu_3 = 0 \tag{4.29}$$

$$\mu\frac{\partial MSE_U(t,\gamma)}{\partial \gamma} - \nu_1 + \nu_2 = 0, \tag{4.30}$$

$$MSE_U(t,\gamma) - MSE_0 K \leq 0 \tag{4.31}$$

$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})} \tag{4.32}$$

$$\mu \geq 0, \nu_1 \geq 0, \nu_2 \geq 0, \nu_3 \geq 0 \tag{4.33}$$

$$\mu(MSE_U(t,\gamma) - MSE_0 K) = 0 \tag{4.34}$$

$$\nu_1(-\gamma) = 0, \nu_2(\gamma - \frac{2}{\lambda_1(\boldsymbol{L})}) = 0, \nu_3(-t) = 0. \tag{4.35}$$

This system seems complex. But it can be simplified as follows. If $MSE(0) = Tr\{\boldsymbol{D}_c\} \leq MSE_0 K$, then $t = 0$ is the solution to the original optimization, which is trivial. Otherwise for $t > 0$, according to (4.35), we have $\nu_3 = 0$. From (4.29), we know $\mu \neq 0$, which leads to $MSE_U(t,\gamma) - MSE_0 K = 0$ according to (4.31). From the constraint $0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})}$, we know $\nu_1 = 0$ and $\nu_2 = 0$ as well. Finally, the KKT system is simplified to

$$\frac{\partial MSE_U(t,\gamma)}{\partial t} + \frac{1}{\mu} = 0 \tag{4.36}$$

$$\frac{\partial MSE_U(t,\gamma)}{\partial \gamma} = 0 \tag{4.37}$$

$$MSE_U(t,\gamma) - MSE_0 K = 0 \tag{4.38}$$

$$0 < \gamma < \frac{2}{\lambda_1(\boldsymbol{L})}, \mu > 0, t > 0. \tag{4.39}$$

Although there are three unknowns t, γ and μ in (4.36)-(4.38), μ can be obtained readily by plugging t and γ into (4.36). Now we only need to solve (4.37) and (4.38) for t and γ with constraint in (4.39).

We can solve the nonlinear system (4.37) and (4.38) by minimizing the following cost function with logarithmic barriers [56]:

$$J(t,\gamma) = \left(\frac{\partial MSE_U(t,\gamma)}{\partial \gamma}\right)^2 + (MSE_U(t,\gamma) - MSE_0 K)^2 - \frac{1}{\tau}\ln t - \frac{1}{\tau}\ln\gamma - \frac{1}{\tau}\ln\left(\frac{2}{\lambda_1(\boldsymbol{L})} - \gamma\right) \tag{4.40}$$

Using gradient descent and Armijo backtracking linear search [56], we can minimize $J(t,\gamma)$ for each choice of τ. Until convergence, the constant τ is increased after each gradient search for t and γ.

After convergence, if $MSE(\lfloor t \rfloor) \leq MSE_0$, the solution is $t^\star = \lfloor t \rfloor$; else if $MSE(\lceil t \rceil) \leq MSE_0$, the solution is $t^\star = \lceil t \rceil$; otherwise, the solution $t^\star$ does not exist.

4.5.3 Other Selections of γ

The previous subsection has presented an optimization of γ with consideration of quantization errors. We will refer to that γ as γ_1. Prior literature has established other choices of γ without any consideration of quantization errors. Two common choices are

$$\gamma_2 = \frac{2}{\lambda_1(\boldsymbol{L}) + \lambda_{K-1}(\boldsymbol{L})} \tag{4.41}$$

$$\gamma_3 = \frac{1}{d_{max}} \tag{4.42}$$

where $d_{max} = \max_{k=1}^{K} |\mathcal{N}_k|$. The choice γ_2 was established in [28] as optimal in the absence of quantization errors. The choice γ_3 was also used in [28] as a simpler option which does not need any further knowledge of $\boldsymbol{L}$.

With γ fixed to γ_2 or γ_3, the corresponding optimal t can be found by choosing the smallest t to satisfy $MSE_j(t,\gamma)/K \leq MSE_0$. The outer loop algorithm for E_0 and B remains the same.

4.6 Simulation Results

For the consensus estimation, the simulation network we consider is same as in Fig. 3.2 in previous chapter for progressive estimation, except that there is no routing tree.

For the simulation, we assume that sensor k observes the data vector $\boldsymbol{x}_k = \boldsymbol{G}_k\boldsymbol{\theta} + \boldsymbol{\omega}_k$, where $\boldsymbol{G}_k$ is the observation matrix associated with sensor k, which is assumed known to sensor k, and $\boldsymbol{\omega}_k$ is white noise with the identity covariance matrix $\boldsymbol{C}_\omega = \boldsymbol{I}$. With this observation model, an original local estimate of $\boldsymbol{\theta}$ at sensor k can be obtained by the best linear unbiased estimate: $\boldsymbol{\theta}_k(0) = (\boldsymbol{G}_k^T\boldsymbol{G}_k)^{-1}\boldsymbol{G}_k^T\boldsymbol{x}_k$. We also have $\boldsymbol{C}_{\theta,k} = (\boldsymbol{G}_k^T\boldsymbol{G}_k)^{-1}$. We will choose $\boldsymbol{G}_k$ for $k = 1, 2, \cdots, K$ randomly. Each $\boldsymbol{G}_k$ is a $N \times M$ real matrix with elements randomly chosen from a Gaussian distribution with zero mean and standard deviation equals to 10.

Each entry of $\boldsymbol{\theta}$ is chosen randomly from $[-1, 1]$. The squared channel gain of channel l from sensor k to its neighboring sensor j is $|h_{k,j,l}|^2 = d_{k,j}^{-\alpha}\rho_{k,j,l}$, where $d_{k,j}$ is the

distance from sensor k to sensor j, $\alpha = 4$ and $\rho_{k,j,l}$ is randomly chosen from an exponential distribution with mean equal to one. We also choose $N = 20$, $M = 10$, $\mu = 1$, $\phi = 1$, $N_0 = 1$, $W_m = 1$ for $m = 1, 2, \cdots, M$.

4.6.1 Effect of γ selection

Fig. 4.3 illustrates the effects of γ on $MSE_1(t, \gamma)$ versus t, where $\gamma_1 = 0.0771$, $\gamma_2 = 0.2163$ and $\gamma_3 = 0.1429$. Here, we assumed $E_0 = 1280$ and $B = 7$ to compute $\boldsymbol{L}$. Recall that $\gamma_3 = 1/d_{max}$ where d_{max} represents a small amount of information from $\boldsymbol{L}$, $\gamma_2 = \frac{2}{\lambda_1(\boldsymbol{L})+\lambda_{K-1}(\boldsymbol{L})}$ which depends much more strongly on $\boldsymbol{L}$, and γ_1 is optimized to minimize t to meet a target MSE where we used $MSE_0 = 0.001$. We have found that $t = 15$ with γ_1. However, with γ_2 and γ_3, the target MSE $MSE_0 = 0.001$ is not achievable with any t.

4.6.2 Effect of E_0 and B

The optimization of E_0 and B corresponds to the outer-loop of the energy planning algorithm for consensus estimation. For each given pair of E_0 and B, the total energy is minimized by the inner-loop of the algorithm. Table 4.1 and Fig. 4.4 illustrate the total energy cost as function of E_0 and B. If E_0 becomes too small, the network loses connectivity between nodes. If B becomes too small, the quantization errors dominate. In either case, the target MSE may become un-achievable even after infinite number of iterations, which corresponds to the case $E_{total} = \infty$.

Corresponding to the curve for consensus estimation in Fig. 3.14 , Fig. 4.5 shows

the optimized B, E_0 and t versus the target MSE.

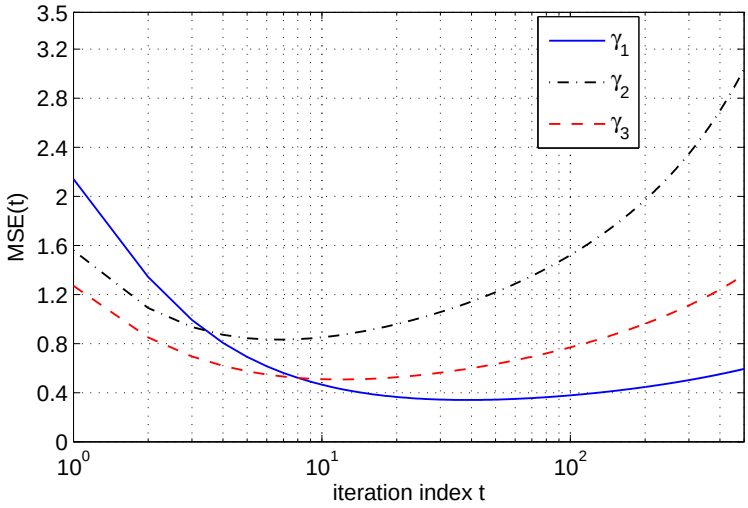

Figure 4.3: $MSE_1(t,\gamma)/K$ of consensus estimation under (4.5) versus the iteration index t, with different γ values: $\gamma_1 = 0.0771$ (optimized with quantization errors), $\gamma_2 = 0.2163$ (optimized without quantization errors) and $\gamma_3 = 0.1429$ (using maximum degree of connectivity), where $E_0 = 1280$, $B = 7$, $MSE_0 = 0.001$.

4.6.3 Consensus versus Progressive on Energy Cost

Fig. 4.6 shows the summation of transmission energy over all sensors with consensus and progressive estimation. The network we consider for both estimation frameworks are the same as in Fig. 3.2 except that there is no routing tree for consensus estimation. Also note that the target MSE for progressive estimation is achieved at the fusion center, but the target MSE for consensus estimation is achieved "on average" at each sensor.

We see that with $p = 1$, progressive estimation with the digital PP algorithm 2 consumes much less sum energy than consensus estimation especially when the target MSE is small. However, with larger p, this energy saving becomes less significant.

Comparing consensus with progressive estimation with the digital PP algorithm 1,

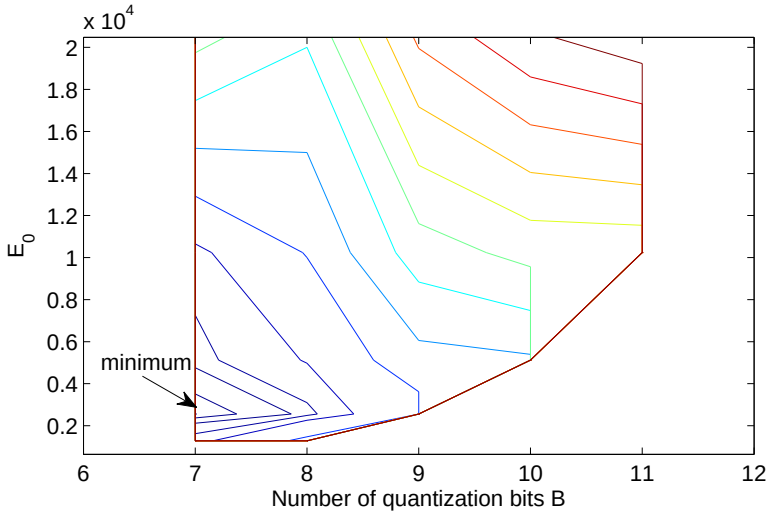

Figure 4.4: The contour of the sum energy required by consensus estimation under (4.5) in terms of E_0 and B. The target MSE is set at $MSE_0 = 0.001$.

we can see at low MSE region, progressive estimation consumes less energy than consensus estimation. However, at high MSE region, progressive estimation can perform worse than the consensus estimation. This actually illustrates that progressive estimation consumes less sum energy than consensus estimation only when a proper energy planning algorithm is applied for progressive estimation.

In Fig. 4.7, we plot the energy of each sensor versus sensor distance to the fusion center. Here, we already order the sensor such that the sensor with smaller index is close to the fusion center. We can easily see that with consensus estimation, the energy distribution is flat over the network. This is conceivable because consensus estimation is a "flat" estimation framework which makes every sensor equally important in the network. However, with progressive estimation, the energy distribution is quite dynamic over the networks. Basically the sensors closer to the fusion center consume more energy than sensors further away to fusion center. This is also intuitively true because progressive estimation is obvious

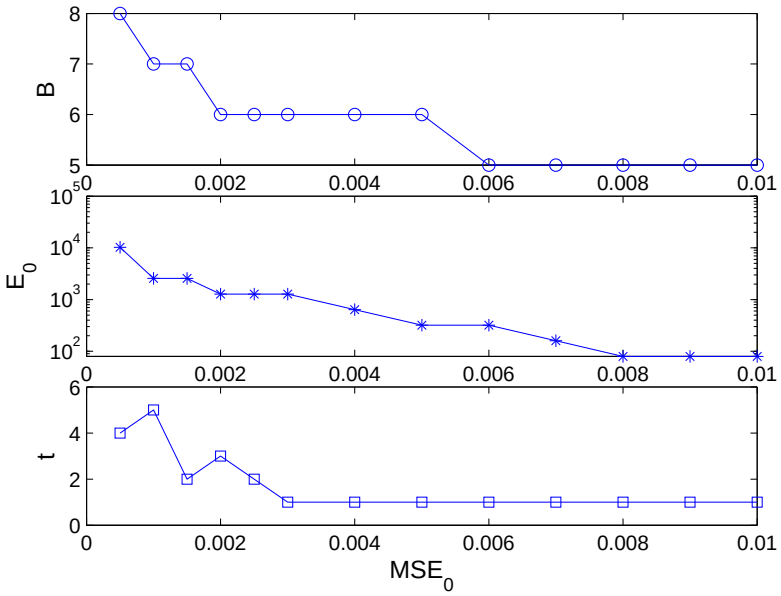

Figure 4.5: The optimal B, E_0 and t required by consensus estimation versus MSE_0.

unequal framework. Those sensors close to the fusion need to refer bits for those far away sensors. Therefore they need more energy to transmit more bits. Moreover, as shown in Fig. 4.7, the peak energy required by progressive estimation (which always occurs near the fusion center) is generally larger than that by consensus estimation.

Fig. 4.8 shows the average number of quantization bits B_k/M allocated for each element at individual sensors. We see that the distribution of B_k/M is almost invariant to the choice of p, and the sensors near the fusion center always uses larger B_k/M . Different from this property, the distribution of the quantization bits generated by the digital PP algorithm 1 shown in previous chapter depends significantly on p, and becomes more uniform when p becomes large.

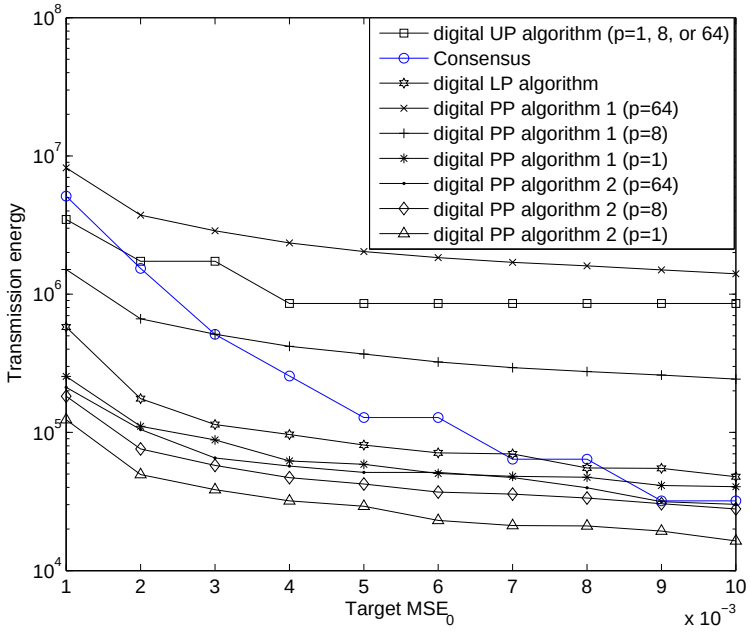

Figure 4.6: Sum energy cost for consensus and progressive estimation.

E_{total}	$B=6$	$B=7$	$B=8$	$B=9$	$B=10$	$B=11$
$E_0=640$	∞	∞	∞	∞	∞	∞
$E_0=1.28\times10^3$	∞	7.68×10^6	2.25×10^7	∞	∞	∞
$E_0=2.56\times10^3$	∞	5.12×10^6	6.14×10^6	1.54×10^7	∞	∞
$E_0=5.12\times10^3$	∞	6.14×10^6	1.02×10^7	2.66×10^7	2.87×10^7	∞
$E_0=1.024\times10^4$	∞	8.19×10^6	2.05×10^7	4.51×10^7	5.32×10^7	5.32×10^7
$E_0=2.048\times10^4$	∞	1.02×10^7	4.1×10^7	8.19×10^7	9.83×10^7	1.06×10^8

Table 4.1: Total transmission energy consumed by the network under different choices of E_0 and B. The MSE target is set $MSE_0=0.001$.

4.7 Summary

This chapter is a comprehensive study of the energy cost on consensus estimation for peer-to-peep multihop sensor networks by jointly taking into account communication and data fusion. Many fundamental communication issues such as transmission energy, quantization bits allocation, and condition for successful packet reception are investigated and incorporated into the design of the consensus estimation. While most previous research works in this area overlook the communication issues and assume data transmission between sensors can be done perfectly. Thus they can focus on studying the convergence conditions and characteristics of data fusion algorithms. Furthermore, we have developed an energy planning algorithm for consensus estimation and illustrated the effects of fusion weights and number of quantization bits, and energy allocation for each sensor on the performance of consensus estimation. The sum energy needed by consensus estimation is shown to be higher than that by progressive estimation with the generalized energy planning algorithm. But the peak energy needed by the former is less than that by the latter.

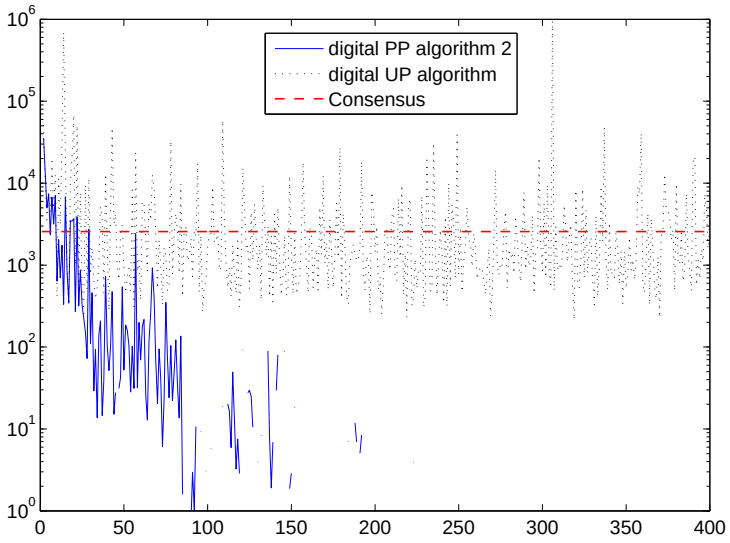

Figure 4.7: Amount of transmission energy (in log scale) consumed by an individual sensor versus the sensor index, where $MSE_0 = 0.001$. For the digital PP algorithm 2, the curve for $p = 1$ is shown here, and the curves for other values of p would differ slightly due to the log scale.

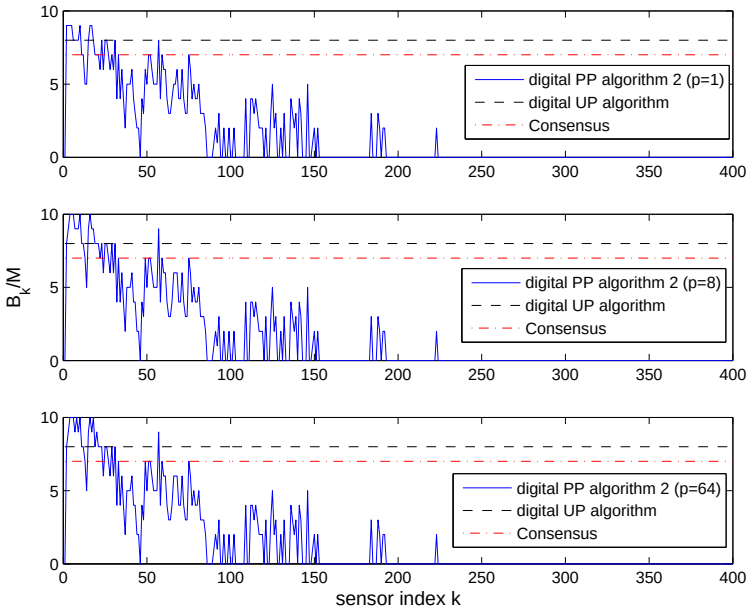

Figure 4.8: Averaged number of quantization bits per element allocated by the digital PP algorithm 2, the digital UP algorithm, and the consensus algorithm, for progressive estimation for an individual sensor versus the sensor index, where $MSE_0 = 0.001$.

Chapter 5

Application Examples

5.1 Camera Network Assisted Football Player Movement Tracking

In this section, we consider an application of progressive estimation, which is using a camera network to track the movement of the players simultaneously in a field game. The game could be a football, basket ball, base ball, hockey, soccer, just to name a few. Hereafter, we consider the football game as an example. As shown in Fig. 5.1, we deploy cameras along the boundary of the football filed. We place a camera every 10 feet. Therefore, we have 36 cameras along the long side, and 16 cameras along the short side. In total, we deployed 104 cameras. Cameras are mounted at certain height such that each of them can cover the whole field. A fusion center, which could be a computer workstation in this application, is deployed at the corner of the field. A two branches routing tree is set up to connect cameras to the fusion center in a multihop fashion. The connections between

cameras are denotes by dashed lines.

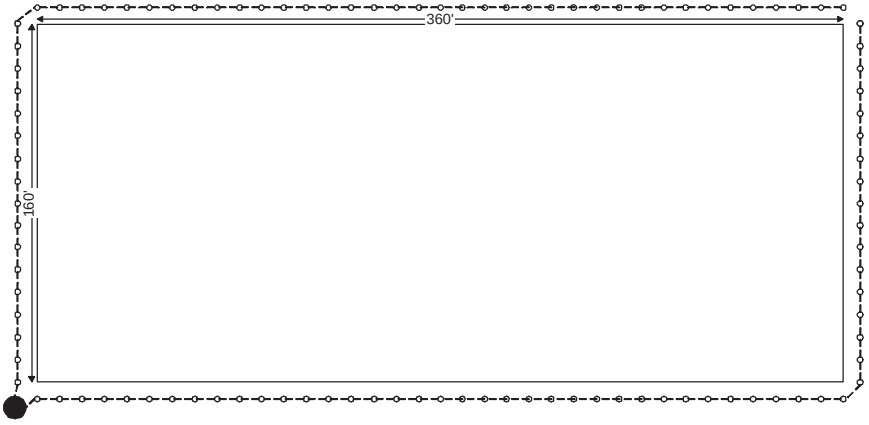

Figure 5.1: The cameras deployment diagram for the football field. The blank little circles are cameras, and the solid big circles denotes the fusion center.

A 3D point in the region of interest is denoted by $\boldsymbol{x} = [x, y, z]^T$. A 2D point on the camera image is denoted by $\boldsymbol{y} = (u, v)$. A camera can usually be modeled by the pinhole, i.e., the relationship between a 3D point and its projection onto the 2D image is given by

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} \tag{5.1}$$

where $\boldsymbol{P}$ is a 2×3 projection matrix, which including the effect of rotation, translation, and effect of camera's intrinsic matrix. In principle, with sufficient control/training points, the projection matrix $\boldsymbol{P}$ can be obtained via the standard least square method. In practice, the problem is called camera calibration. There are already many mature techniques in the literature available to obtain the matrix $\boldsymbol{P}$, see [57] and the references therein.

In this football players tracking problem, we could further simplify the projection model, because all players move on the same ground plain. Without loss of generality, we can choose $z = 0$.

We denote the coordinates of the jth target at time t on the 2D ground as $\boldsymbol{x}_j(t) = [x_j(t), y_j(t)]^T$. Based on the above model, the projection of $\boldsymbol{x}_j(t)$ onto the kth camera's image plane is given by

$$\boldsymbol{y}_{k,j} = \boldsymbol{P}_k \boldsymbol{x}_j(t) + \boldsymbol{\omega}_k \tag{5.2}$$

where $\boldsymbol{P}_k$ is now a 2×2 matrix, which is pre-calibrated thus known to the fusion center. $\boldsymbol{\omega}_k$ is the observation noise vector due to the calibration error of camera k. We assume the noise has zero mean. We also assume we know its covariance matrix $\boldsymbol{C}_k$, which is feasible to get during the calibration process. By left-multiplying $\boldsymbol{C}_k^{-1/2}$ on both sides of the above equation, we can whiten the noise. Therefore, without loss of generality, we could assume the noise $\boldsymbol{\omega}_k$ is white with zero mean and unit variance.

With the above linear data model, together with the generalized progressive estimation we developed, we could estimate the unknown vectors $\boldsymbol{x}_j(t)$. Repeating the process for each snap shot t, we can track the trajectory of each single player.

In the simulation, we select the error margin $MSE_0 = 0.001$ square feet for the tracking performance requirement. We choose $J = 23$, i.e., we track 22 players on both sides and the movement of the ball simultaneously. While, for convenience of plot of movement trajectories, we only plot the movements of the offense team in total of 11 players in Fig. 5.2. The Fig. shows that the estimated trajectories are very close to the actual movements of the players. To show the performance more clearly, we choose two players and enlarge the plot of their actual and estimated movements. The small plot at the upper left corner of the figure shows the estimated and actual movements of a wide receiver in one offense possession. The small plot at the lower right corner shows the estimated and actual movements of

the quarter back, whose actual trajectory is models as a Brownian motion. One should notice that even the randomness of the brownian motion will not degrade the estimation performance.

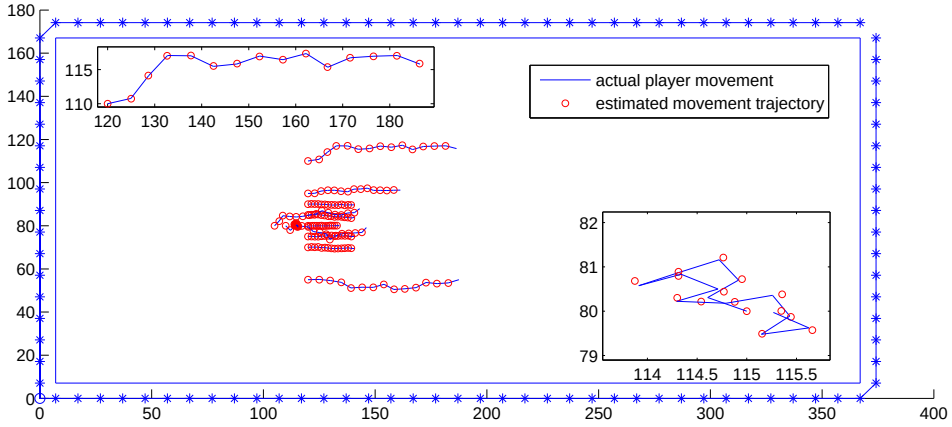

Figure 5.2: The actual and tracked offense players movements in one possession.

5.2 Sensor Network Assisted Targets Tracking

In this section, we consider another application of progressive estimation, which is to use a sensor network for target tracking in the region of interest. For example, we want to monitor the movement of several unidentified aircrafts in a homeland security event. Traditionally, radar signal processing techniques are adopted for this application. We intend to replace the expensive radar stations by some inexpensive networked sensors, while still guarantee the tracking performance.

We deploy in total K sensors and a fusion center in the region of interest, with a routing tree from sensors to the fusion center pre-established. We assume sensors' positions

(x_k, y_k, z_k) are all fixed, and known to the fusion center. Through detection techniques, we know there are in total J targets moving in the region of interest. The task now is to track the trajectories of those moving targets in the region. We denote the coordinates of the jth target at time t as $[x_{0,j}(t), y_{0,j}(t), z_{0,j}(t)]^T$. We assume, weather actively or passively, a signal at one certain frequency is periodically emitted from these targets. This signal is then received by each sensor individually. The received signal strength embeds the information of targets locations. By utilizing received signals of all sensors collaboratively, the estimation of the targets locations at each time t can be obtained at the fusion center. By repeating the process at each time slot t, we could obtain the trajectories of all the moving targets.

For the signal power attenuation in wireless environment, we introduce the following isotropic signal attenuation model. The received signal power at a receiver with distance d to the transmitter is given by

$$P = GP_0 \left(\frac{d}{d_0}\right)^{-\alpha} \tag{5.3}$$

where G is the processing gain of the receiver. $d_k(t)$ is the Euclidean distance between the transmitter and the receiver. P_0 is the emitted power of the target measured at a reference distance d_0, and α is the signal power attenuation component, which both can be found via experimental data. Normally, for free space, we know $\alpha = 2$. However, in an wireless environment with intensive reflections and diffusions, α can be 5 or even larger.

We denote the measured signal power at sensor k at time t as $s_k(t)$. We assume $s_k(t)$ is corrupted by a measurement noise $n_k(t)$, i.e.,

$$s_k(t) = G_k P_0 d_0^{\alpha} \sum_{j=1}^{J} d_{k,j}^{-\alpha}(t) + n_k(t) \tag{5.4}$$

Here, $d_{k,j}(t) = \sqrt{(x_k - x_{0,j}(t))^2 + (y_k - y_{0,j}(t))^2 + (z_k - z_{0,j}(t))^2}$, which is the distance from the jth target to sensor k at time t. We assume the noise is stationary with zero mean and constant variance σ_k^2. Otherwise, we do not assume any further knowledge about the noise distribution.

The above data model is a nonlinear model. We linearize it based on the Taylors expansion as below

$$s_k(t) \approx \sum_{j=1}^{J} \frac{G_k P_0 d_0^\alpha}{d_{k,j}^\alpha(t-1)} + \sum_{j=1}^{J} \frac{\partial s_k(t)}{\partial x_{0,j}}|_{x_{0,j}(t-1)} \left(x_{0,j}(t) - x_{0,j}(t-1)\right) \tag{5.5}$$
$$+ \sum_{j=1}^{J} \frac{\partial s_k(t)}{\partial y_{0,j}}|_{y_{0,j}(t-1)} \left(y_{0,j}(t) - y_{0,j}(t-1)\right) + \sum_{j=1}^{J} \frac{\partial s_k(t)}{\partial z_{0,j}}|_{z_{0,j}(t-1)} \left(z_{0,j}(t) - z_{0,j}(t-1)\right)$$

It is easy to verify that

$$\frac{\partial s_k(t)}{\partial x_{0,j}}|_{x_{0,j}(t-1)} = \alpha G_k P_0 d_0^\alpha d_k(t-1)^{-\alpha-2} \left(x_k - x_{0,j}(t-1)\right), \tag{5.6}$$

$$\frac{\partial s_k(t)}{\partial y_{0,j}}|_{y_{0,j}(t-1)} = \alpha G_k P_0 d_0^\alpha d_k(t-1)^{-\alpha-2} \left(y_k - y_{0,j}(t-1)\right), \tag{5.7}$$

and

$$\frac{\partial s_k(t)}{\partial z_{0,j}}|_{z_{0,j}(t-1)} = \alpha G_k P_0 d_0^\alpha d_k(t-1)^{-\alpha-2} \left(z_k - z_{0,j}(t-1)\right). \tag{5.8}$$

In summary, an approximated linear data model can be established as follows:

$$\bar{s}_k(t) = \boldsymbol{g}_k(t)\boldsymbol{l}(t) + \bar{\omega}_k \tag{5.9}$$

Here, the essential observation sample $\bar{s}_k(t)$ is given by $\bar{s}_k(t) = \frac{1}{\sigma_k}\left(s_k(t) - \sum_{j=1}^{J} \frac{G_k P_0 d_0^\alpha}{d_{k,j}^\alpha(t-1)}\right) + \boldsymbol{g}_k(t)\boldsymbol{l}(t-1)$. $\boldsymbol{l}(t)$ is the vector contains all parameters of target locations, i.e., $\boldsymbol{l}(t) = [\cdots, x_{0,j}(t), y_{0,j}(t), z_{0,j}(t), \cdots]^T$. $\bar{\omega}_k$ is the normalized noise with zero mean and unit variance. $\boldsymbol{g}_k(t)$ is a $1\times 3J$ observation vector $[\cdots, \frac{1}{\sigma_k}\frac{\partial s_k(t)}{\partial x_{0,j}}|_{x_{0,j}(t-1)}, \frac{1}{\sigma_k}\frac{\partial s_k(t)}{\partial y_{0,j}}|_{y_{0,j}(t-1)}, \frac{\partial s_k(t)}{\partial z_{0,j}}|_{z_{0,j}(t-1)}, \cdots]$.

With the above linear model, together with the generalized progressive estimation we developed, we could estimate the unknown vector $\boldsymbol{l}_t$ repeatedly for each time t. We consider the region of interest to be a cubic area of $20D \times 20D \times 20D$, where D is the pre-specified distance normalizing factor. The sensors are deployed on the $x-y$ plane as plotted in Fig. 3.2. We plan to utilize the 2-d deployed sensors to track a moving target in the 3-d region of interest.

In simulation, we select the error margin $MSE_0 = 0.01$ for the tracking performance requirement. Without loss of generality, we assume the reference distance $d_0 = D$ for the wireless signal attenuation model (5.3). We assume $\alpha = 4$, and $G_k P_0 d_0^{\alpha} = 1000$. We assume the sensor observation noise variance $\sigma^2 = 10^{-6}$ homogeneously for all sensors. Fig. 5.3 demonstrate that the estimated target trajectory are very close to the actual target movement trajectory.

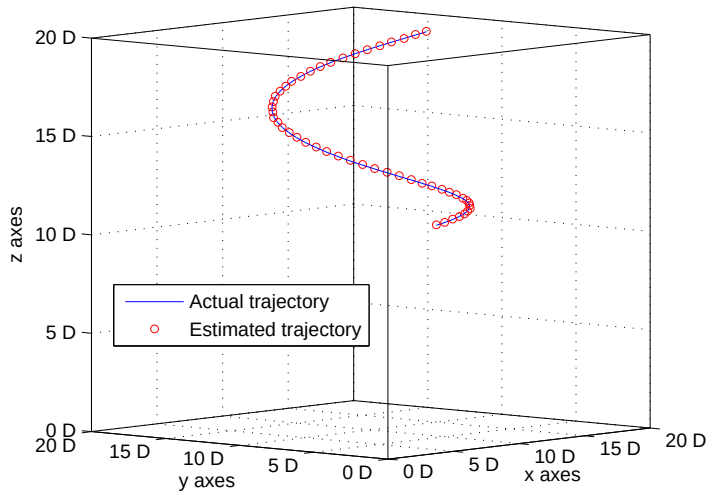

Figure 5.3: The actual target trajectory and the estimated trajectory.

Chapter 6

Conclusion

The first contribution of this thesis is the progressive data fusion framework, which is very different to the traditional single-hop and consensus data fusion framework. Progressive data fusion extend single-hop data fusion to multi-hop data fusion where information transmission only taking place between adjacent neighboring sensors such that significant energy saving can be achieved, especially in large scale sensor networks. Unlike the consensus data fusion whose convergence is asymptotic due to its iterative nature, the performance of progressive data fusion can be guaranteed in a predetermined time window. Also, in terms of sum of energy consumed over all sensors, progressive data fusion is the best among the three.

The contributions in Chapter 3 on the topic of energy planning for progressive data fusion include: 1) a comparison of digital communication and analog communication in terms of transmission energy and transmission errors; 2) an energy planning algorithm for progressive estimation in multihop networks using digital communication; 3) an energy

planning algorithm for progressive estimation in multihop networks using analog communication; 4) the use of L_p norm for both energy planning algorithms (with L_1 norm measuring the sum energy and L_∞ norm measuring the peak energy); 5) We show that digital communication is more efficient in transmission energy than analog communication unless the available transmission time-bandwidth product (i.e., $\mathcal{TW}$ or its integer part L) for each link and each observation sample is very limited. This fundamental observation provides a useful perspective of the prior works shown in [18] and [19] where analog communication is assumed between sensors and the fusion center; 6) With the time-bandwidth product L for transmission of a message, one can use L parallel sub-channels for digital transmission of the same message, the importance of which has been somehow overlooked in the literature on the modelling and analysis of sensor network problems; 7) Our proposed digital transmission energy planning algorithm 1 and algorithm 2 are applicable to any L (but no less than one). This is another useful extension from [13] and [24] where the freedom of L is not utilized.

The contributions in Chapter 4 on the topic of energy planning for consensus data fusion include: 1) We study the consensus estimation for sensor networks by taking into account many fundamental wireless communication issues such as transmission energy optimization, quantization bits allocation, condition of successful packet reception. In the literature [26]-[45], wireless communications are assumed to be done perfectly and those issues are overlooked. 2) In this paper, we focus on minimizing the energy cost for the whole network to achieve a pre-given MSE threshold, while most previous papers in this area study the convergence conditions and characteristics of different estimation schemes. 3)

We also compare the energy cost of consensus estimation with that of progressive estimation under the same sensor constellation and the same target performance. Consensus estimation is shown to consume more sum energy than progressive estimation although the peak energy for individual sensors by the former is less than that by the latter.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magzine*, vol. 40, pp. 102–114, Aug. 2002.

[2] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad hoc sensor networks," *J. Computer Commun. Sensor Networks*, vol. 29, pp. 413–420, Feb. 2006.

[3] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, pp. 924–935, Sept. 2002.

[4] W. Ye, J. Heidemann, and D. Estrin, "Data gathering algorithms in sensor networks using energy metrics," in *Proc. 21st Annual Joint Conf. IEEE Computer Commun. Societies*, New York, Jun. 2002.

[5] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Commun. Mag.*, vol. 11, pp. 6–28, Dec. 2004.

[6] D. A. Castanon and D. Teneketzis, "Distributed estimation algorithms for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 30, pp. 418–425, May 1985.

[7] Z. Chair and P. K. Varshney, "Distributed bayesian hypothesis testing with distributed data fusion," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 695–699, Sep.-Oct. 1988.

[8] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part i - fundamentals," *Proc. IEEE*, vol. 85, pp. 54–632, Jan. 1997.

[9] Z.-Q. Luo and J. N. Tsitsiklis, "Data fusion with minimal communication," *IEEE Trans. Inf. Theory*, vol. 40, pp. 1551–1563, Sep. 1994.

[10] T. Berger, Z. Zhang, and H. Viswanathan, "The ceo problem," *IEEE Transactions on Information Theory*, vol. 42, pp. 887–902, May 1996.

[11] Z.-Q. Luo, "Universal decentralized estimation in a bandwidth constrained sensor networks," *IEEE Trans. on Information Theory*, vol. 51, pp. 2210–2219, Jun. 2005.

[12] J.-J. Xiao and Z.-Q. Luo, "Decentralized estimation in an inhomogeneous sensing environment," *IEEE Trans. on Information Theory*, vol. 51, pp. 3564–3575, Oct. 2005.

[13] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Trans. on Signal Processing*, vol. 54, pp. 413–422, Feb. 2006.

[14] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, part i: Gaussian pdf," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1131–1143, Mar. 2006.

[15] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, part ii: Unknown pdf," *IEEE Transactions on Signal Processing*, vol. 54, pp. 2784–2796, Jul. 2006.

[16] P. Venkitasubramaniam, L. Tong, and A. Swami, "Quantization for maximin are in distributed estimation," *IEEE Transactions on Signal Processing*, vol. 55, pp. 3596–3605, Jul. 2007.

[17] I. D. Schizas, G. B. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4284–4299, Aug. 2007.

[18] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. Goldsmith, "Linear coherent decentralized estimation," *IEEE Transactions on Signal Processing*, vol. 56, pp. 757–770, Feb. 2008.

[19] S. Cui, J.-J. Xiao, A. G. Goldsmith, Z.-Q. Luo, and H. V. Poor, "Estimation diversity and power efficiency in distributed sensing," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4683–4695, Sept. 2007.

[20] J. Li and G. Alregib, "Distributed estimation in energy-constrained wireless sensor networks," *IEEE Trans. on Signal Processing.*, vol. 57, pp. 3822–3832, Oct. 2009.

[21] L. Zhang, X. Zhang, T. Ho, and T. Dikaliotis, "Progressive distributed estimation over noisy channels in wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009.

[22] J. Fang and H. Li, "Power constrained distributed estimation with cluster-based sensor collaborations," *IEEE Trans. on Wireless Comm.*, vol. 8, pp. 3822–3832, Jul. 2009.

[23] Y. Hua and Y. Huang, "Progressive estimation and detection," in *Workshop on Sensor Signal and Information Processing (SenSIP)*, Sedona, AR, May 2008.

[24] Y. Huang and Y. Hua, "Multihop progressive decentralized estimation in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 14, pp. 1004–1007, Dec. 2007.

[25] Y. Huang and Y. Hua, "Energy planning for progressive estimation in multihop sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, pp. 4052–4065, Oct. 2009.

[26] M. H. DeGroot, "Reaching a consensus," *Journal of American Statistical Association*, vol. 69, pp. 118–121, 1974.

[27] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of The IEEE*, vol. 95, pp. 215–233, Jan. 2007.

[28] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, Sept. 2004.

[29] D. Scherber and H. C. Papadopoulos, "Distributed computation of averages over ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 776–787, Apr. 2007.

[30] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 798–808, Apr. 2005.

[31] S. Barbarossa and G. Scutari, "Decentralized maximum-likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems," *IEEE Trans. on Signal Processing*, vol. 55, pp. 3456–3470, Jul. 2007.

[32] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Processing*, vol. 56, pp. 3122–3136, Jul. 2008.

[33] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links - part i: distributed estimation of deterministic signals," *IEEE Trans. Signal Processing*, vol. 56, pp. 350–364, Jan. 2008.

[34] I. D. Schizas, G. B. Giannakis, S. D. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links - part ii: distributed estimation and smoothing of random signals," *IEEE Trans. Signal Processing*, vol. 56, pp. 1650–1666, Apr. 2008.

[35] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Information Theory*, vol. 52, pp. 4753–4766, Nov. 2006.

[36] S. Boyd, A. Ghosh, B. Prabhakar, , and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Information Theory*, vol. 52, pp. 2508–2530, Jun. 2006.

[37] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time delays," *IEEE Trans. Automatic Control*, vol. 49, pp. 1520–1533, Sept. 2004.

[38] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automatic Control*, vol. 50, pp. 655–661, Sept. 2005.

[39] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2748–2761, Jul. 2009.

[40] T. C. Aysal and K. E. Barner, "On the convergence of perturbed non-stationary consensus algorithms," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009.

[41] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Transactions on Signal Processing*, vol. 56, pp. 4905–4918, Oct. 2008.

[42] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, pp. 33–46, 2007.

[43] F. Fagnani P. Frasca, R. Carli and S. Zampieri, "Average consensus on networks with quantized communication," *International Journal of Robust and Nonlinear Control*, vol. 19, pp. 1787–1816, 2009.

[44] R. D. Nowak M. G. Rabbat and J. A. Bucklew, "Generalized consensus computation in networked system with erasure links," in *IEEE Workshop on Signal Processing Advances in Wireless Communications*, New York, Jun. 2005.

[45] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, pp. 355–369, Jan. 2009.

[46] F. C. Harris Jr., *Steiner minimal trees: an introduction, parallel computation, and future work: Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 1998.

[47] S. Haykin, *An Introduction to Analog and Digital Communications*, New York: Wiley, 1994.

[48] Jr. G. D. Forney and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *Transactions on Information Theory*, vol. 44, pp. 2384–2415, Oct. 1998.

[49] S. T. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, Mar. 1982.

[50] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

[51] S. C. Draper and G. W. Wornell, "Side information aware coding strategies for sensor networks," *IEEE Journal on selected areas in Communications*, vol. 22, pp. 96–1067, Aug. 2004.

[52] M. Gastpar and M. Vetterli, "Power, spatial-temporal bandwidth, and distortion in large sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 745–754, Apr. 2005.

[53] M. Gastpar, "Uncoded transmission is exactly optimal for a simple gausssian sensor network," *IEEE Transactions on Information Theory*, vol. 54, pp. 5247–5251, Nov. 2008.

[54] S. Cui, J.-J. Xiao, A. J. Goldsmith, Z.-Q. Luo, and H. V. Poor, "Energy-efficient joint estimation in sensor networks: analog vs. digital," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, Apr. 2005.

[55] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[56] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[57] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, Nov. 2000.