DELAY-CONSTRAINED AND ENERGY-EFFICIENT DISTRIBUTED ALGORITHMS FOR COMPUTATION-INTENSIVE APPLICATIONS IN WSNS

LIN WANYU

M.Phil

The Hong Kong Polytechnic University

2015

THE HONG KONG POLYTECHNIC UNIVERSITY
DEPARTMENT OF COMPUTING

DELAY-CONSTRAINED AND ENERGY-EFFICIENT DISTRIBUTED ALGORITHMS FOR COMPUTATION-INTENSIVE APPLICATIONS IN WSNS

Lin Wanyu

A thesis submitted in partial fulfillment
of the requirements for the degree of Master of Philosophy

January 2015

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis entitled "Delay-constrained and energy-efficient distributed algorithms for computation-intensive applications in WSNs" is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________ (Signed)

Liu Wanyu (Name of Student)

Abstract

In recent years, research on wireless sensor networks(WSNs) has gradually spread from the traditional applications such as environmental monitoring to more domain-specific applications which are computation intensive due to large amount of collected physical data and complexity of the computation, such as structural health monitoring (SHM), volcano tomography, smart grid. However, for these applications, due to severe limitations of energy and bandwidth, it is necessary to utilize the computation capability of sensors and allow them to process raw signals within the network, and only transmit processed information. This implies a new revolution of designing energy-efficient distributed computing architecture for computation-intensive applications in WSNs. The distributed computing systems are scalable in the sense that all the computational and networking capacities scattered across the network could be utilized in a cooperative and distributed (not master-to-slave) manner.

Considering the data-intensive and computation-intensive properties for some domain-specific applications, several unique challenging issues arise. Firstly, those algorithms designed by domain experts usually only consider the design aspects from domains such as accuracy, and they are usually sophisticated signal processing algorithms. Most algorithms are associated with complex computations such as large matrix inversion, matrix multiplication in which matrices are constructed by the raw data from different sensors. Therefore, designing a distributed version to perform matrix operations when considering the severe constraint of wireless network resources (bandwidth, energy, computing capability, memory, etc) is difficult.

In this research, we propose a framework focusing on how to implement sophisticated processing of intensive physical information within a network. We focus on the design of distributed estimation algorithms for least squares estimation. Recent years, researchers have proposed a wide range of strategies for distributed least squares estimation. However,

each strategy has its own design objectives and applications scenarios. No guided schemes exists for current practical usage, making it difficult to evaluate their relative effectiveness and performance. Thus, we propose a 3 dimensional framework which provides a basis for designing, analyzing and evaluating strategies to address parameters estimation issues using least squares estimation algorithms in wireless sensor networks. In the 3D framework, we propose three design aspects of designing distributed least squares estimation, and then we study the existing works from the design aspects and then discuss their advantages and disadvantages, respectively.

Finally, based on our proposed framework, we wish to conserve energy by minimizing communication with our new design, constraints on communication delays will also need to be satisfied. Thus, we propose E^3, a new distributed algorithm specifically designed to guarantee the precision of least squares estimation in sensor networks, with the objective of minimizing the energy consumption incurred during communication, while observing constraints on application-specific communication delays. To evaluate the performance of our proposed framework and algorithms, we conduct simulations and structural damage detection experiments in a real environment to do test. Compared to previous works, we show that E^3 maintains the same level of estimation precision while incurring much lower energy costs. Finally, we address that our 3D framework is the first work which can facilitate the design, classification and evaluation of the current distributed least squares estimation strategies in sensor networks.

Publications

Journal Paper

1. **Wanyu Lin**, Jiannong Cao, and Xuefeng Liu, "*Distributed Least Squares Estimation Algorithms Design for Sensor Networks*", submitted to Ad Hoc & Sensor Wireless Networks

Conference Paper

1. **Wanyu Lin**, Jiannong Cao, and Xuefeng Liu, "*E^3: Towards Energy-Efficient Distributed Least Squares Estimation in Sensor Networks*", accepted by IEEE International Symposium on Quality of Service (IWQoS), 2014

Acknowledgements

First of all, I would like to express my gratitude to all those who helped me during my MPhil. program study. My deepest gratitude goes first and foremost to my supervisor, Prof. Jiannong Cao, for his systematic guidance, illuminating advice and encouragement in my research work. His broad knowledge, keen insight, and enormous enthusiasm on the research inspire me a lot and encourage me to keep going on in my study. Especially, he always trained me to be a qualified researcher, not only about the research technologies and presentation skills, but also rigorous work attitude. I appreciate all these and will be great benefited from him not only in my future works but also in my life.

I would like to thank Dr. Xuefeng Liu. He discussed with me from time to time, taught me how to do research and gave me several constructive suggestions in my research and great help in the paper writing.

I would also like to thank my parents, my sisters and little brother. They always encourage and support me when I encounter difficulties in different aspects. Their love is the most powerful motivation I can make progress in my work and my life.

I want to thank my colleagues Dr. Weiping Zhu, Dr. Lei Yang, Dr. Tao Li, Ms. Yin Yuan, Mr. Rui Liu, Mr. Junhao Zheng, Mr. Yang Liu, Ms. Lin Xu, Mr. Weiqing Liu, Mr. Zongjian He, Dr. Peng Guo, Mr. Linchuan Xu and all other members of our research group that I cannot enumerate here. Thank you for your help and support in these years. We learn from each other, share our joyfulness and sadness, and have a unforgettable memory together. I wish all of you a brilliant future and a happy life.

Last but not the least, I also want to thank my roommates Ms. Yiqing Cai, Ms. Chengyao Chen. I am lucky to meet you in Hong Kong and appreciate your kindly help in my life. Thank you for your accompany.

Table of Contents

Abstract **i**

Publications **iii**

Acknowledgements **v**

Table of Contents **vii**

List of Figures **ix**

List of Tables **xi**

List of Abbreviations **xiii**

1 Introduction **1**
- 1.1 Wireless Sensor Networks 1
- 1.2 Data-intensive and Computation-intensive Properties of Specific Applications 3
- 1.3 Least Squares Estimations 4
- 1.4 Motivations of Our Work 4
- 1.5 Contributions of the Thesis 5
- 1.6 Organization of the Thesis 6

2 Literature Review **9**
- 2.1 Existing Works on signal-processing algorithms in the applications of WSNs 9
  - 2.1.1 Computation Lightweight SPA 9
  - 2.1.2 Computation Intensive SPA 10
- 2.2 Existing Works on distributed least squares estimation algorithms in WSNs 13

3 3D Framework for Distributed Least Squares Estimation Algorithms Design for Sensor Networks **15**
- 3.1 Overview 15
- 3.2 System Model 17
- 3.3 Design Aspects of Our 3 Dimensional Framework 21
  - 3.3.1 Aggregation Structure 22
  - 3.3.2 Performance Metric 24

3.3.3 Communication Pattern . . . 25
3.4 3D Design Framework . . . 25
3.5 Summary . . . 29

4 E^3: Towards Energy-Efficient Distributed Least Squares Estimation in Sensor Networks 31
4.1 Overview . . . 31
4.2 System Model and Problem Formulation . . . 33
4.2.1 System Model . . . 33
4.2.2 Problem Formulation . . . 34
4.3 E^3: Distributed Least Squares Estimation . . . 37
4.3.1 Incremental recursive least squares estimation . . . 37
4.3.2 Distributed least squares estimation: a simple strategy as the starting point . . . 38
4.3.3 E^3: Our distributed algorithm . . . 41
4.4 Performance Evaluation . . . 44
4.4.1 Simulation . . . 44
4.4.2 Experiments in a building . . . 47
4.5 Summary . . . 53

5 Conclusions and Suggestions for Future Research 55
5.1 Conclusions . . . 55
5.2 Suggestions for Future Research . . . 56
5.2.1 Distributed Computing Architecture . . . 57

References 59

List of Figures

1.1 Using cables to collect data and centralized processing to wireless data collection and distributed computation 2

1.2 An outline of the contributions in this thesis 7

2.1 Algorithms with different levels of collaboration(a)at decision level(b)at feature level(c)at data level 11

2.2 Cluster-based modal parameter estimation 12

3.1 A wireless sensor network: an example of our system model. 18

3.2 Computing the least squares estimation at the sink node in a centralized manner, while measurement data is being transmitted in a shortest path tree to minimize the transmission cost. 20

3.3 Different aggregation structures for sophisticated algorithms 23

3.4 3D design framework. Each axis represents one design aspect and contains a range of properties. 26

4.1 Incremental recursive least squares estimation along a Hamiltonian path [SL06]. 38

4.2 Distributed least squares estimation: a simple strategy. 40

4.3 Distributed least squares estimation using E^3. 43

4.4 The transmission cost with various network densities. 46

4.5 The transmission cost with various network sizes. 47

4.6 The building and measurement locations (a) The building (b) 20 measurement locations. 49

4.7 Experiment setup (a) Wired and wireless systems deployed at location 17 in the building (b) Top: a SHM mote deployed near the window as the collector node; Bottom: a gateway mote connected to a laptop computer. 50

4.8 (a) The network topology of collector nodes (b) The shortest path tree (with root 20) and (c) The minimum leaves tree (with root 20). 52

List of Tables

4.1 Theoretical data amount and the actual maximum hop count in different strategies. . . . 51

4.2 Comparing accuracy of our approach to the centralized LSE. It can be seen that the identified natural frequencies of our approach have less than 1% identification error. 52

List of Abbreviations

SHM: Structural Health Monitoring

LSE: Least Squares Estimation

WSN: Wireless Sensor Network

SPA: Signal Processing Algorithm

Chapter 1

Introduction

This research aims to investigate the issues and design novel distributed computation algorithms for data-intensive and computation-intensive applications in wireless sensor network(WSNs) . In this chapter, we first describe the background knowledge of WSNs and domain-specific applications with data-intensive and computation-intensive properties in Section 1.1 and Section 1.2 respectively. Then, we introduce least squaress estimation, which is one of the most fundamental signal processing algorithms in domain-specific applications in Section 1.3. After that, we explain the motivation of our work in Section 1.4. In Section 1.5, we summarize the main contributions of this thesis. Finally, we outline the organization of this thesis in Section 1.6.

1.1 Wireless Sensor Networks

In the last decade, advances in micro-electro-mechanical systems (MEMS) technology, digital electronics, and wireless communications have enabled the development of low-cost, low-power, multi-functional smart sensor nodes. These smart sensors formed Wireless Sensor Networks(WSNs) which consist of sensing, data processing, and communicating components, have been used in many science and engineering areas. Different from traditional network which uses cables to collect data from a number of sensors and a powerful back-end server to extract information, WSNs represent a new paradigm which relies on massively distributed collections of smart sensors embedded in the physical world,such as SHM as Fig. 1.1 shows. These smart sensors work in unattended way to gather data from physical

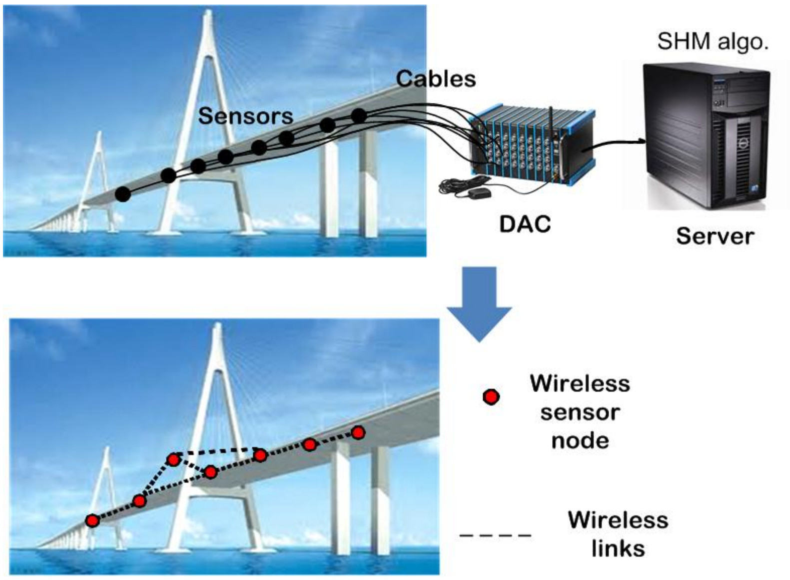


Fig. 1.1: Using cables to collect data and centralized processing to wireless data collection and distributed computation

worlds, exchange information through wireless or wired networks, and implement signal processing in a collaborative manner. Compared with traditional systems, the advantages of low cost and ease of maintenance, the scalability, and the ability to take up close look at phenomena make the WSNs a revolutionary paradigm which is able to help to obtain a deeper understanding of the environment and, ultimately, enhancing our ability to design and control these complex systems. WSNs have envisioned to be used in numerous application domains including environmental monitoring [MCP+02] [KPKK07] [YWM05], intelligent transportation system, structural health monitoring (SHM) of large buildings and bridges [BA], state-estimation in smart grid [GEAdlVJGQ11], battle-field surveillance, biomedical detection [CLW12] and human health monitoring.

1.2 Data-intensive and Computation-intensive Properties of Specific Applications

In the last twenty years, the practical usage of WSNs is surprisingly still limited to a narrow range of application areas such as monitoring fire in a forest, temperature in a building, etc. [MCP+02] [KPKK07] [YWM05]. These applications are characterized as having low data rates and with light-weight computations. In particular, each smart sensor usually samples and transmits data once every few minutes, and only implements (if there does exist any) very simple in-network processing which is typically limited to spatial or temporal aggregation of the collected data (such as calculating the average or maximum sensor value over some region).

On the contrary, there are a large number of potential application domains of WSNs which are data-intensive, and require sophisticated processing of the collected physical information. Among the examples are SHM [LL06], volcano tomography [SSX+], state estimation in smart grid [GEAdlVJGQ11], wireless camera networks [MPK08], biomedical monitoring [CLW12], and fault diagnosis of machines.

An application becomes data-intensive mainly because of its high-fidelity sampling as well the large number of nodes. Taking SHM for instance, to detect possible structural damage, each accelerator deployed in a structure needs to sample in the range of 16-24 bit at 200-1000Hz [DFPS96]. The measurements sampled from each sensor are hence no longer binary or static data, but highly dynamic time series. With such high-fidelity sampling, a SHM system including hundreds of sensors generates a large amount of data in a short period of time. Another example is smart grid. In smart grid, precise real-time estimation of the state variables is required to maintain the reliable operation. Estimation of these state variables requires the data from hundreds of thousands sensors and meters distributed across city-wide areas. The amount of collected data thus becomes huge.

1.3 Least Squares Estimations

Least squares estimation is a standard approach to compute the approximate solution of sets of equations in which there are more equations than unknowns. The domain-specific problems with data-intensive and computation-intensive features can usually be formulated as a large linear least squares system based on measurements from sensors. These measurements may contain noise, thus redundant measurements that are used to reduce the effects of error are often sampled to form an over-determined system:

$$H \times \omega^0 = y \tag{1.1}$$

where $H \in \mathbb{R}^{(m \times n)}$ $(m \geq n)$ is the regression matrix, the scalar measurements $y \in \mathbb{R}^m$ and the unknown parameters $\omega^0 \in \mathbb{R}^n$.

In wireless sensor networks, the primary objective is to collectively estimate n unknown parameters which form a $n \times 1$ column vector, denoted by ω^0 based on the measurements (H, y) from all sensors.

Due to the data-intensive nature of the domain-specific applications, the formulated least square systems are always quasi-over-determined.

We use structure damage detection as our application scenario to illustrate what does least squares estimation for. In structural health monitoring (SHM) domain, H and y are composed of pre-processed information of raw vibration data measured in each sensor. The unknown parameters are used to obtain natural frequencies, damping ratio or mode shapes of the structure[Avi01].

1.4 Motivations of Our Work

For the data-intensive and computation-intensive applications, existing systems with data collection and then centralized computation architecture cannot be applied. Due to severe limitations of energy and bandwidth (particularly for those using wireless) at current, battery-powered smart sensors, it is virtually impossible to collect raw, real-time data from a large-scale (e.g., hundreds to thousands), dense sensor network. To solve this problem, it is necessary to utilize the computation capability of smart sensors and allow them to process raw signals within the network, and only transmit processed information. This implies a new revolution of designing energy-efficient distributed computing architecture for data-intensive and computation-intensive applications in WSNs. The distributed systems are scalable in the sense that all the computational and networking capacities scattered across the network could be utilized in a cooperative and distributed (not master-to-slave) manner.

In this thesis, we will analyze aforementioned problems in details and propose corresponding solutions for them.

1.5 Contributions of the Thesis

The contributions of this thesis mainly lie in designing novel distributed algorithms for least squares estimation in wireless sensor networks. As illustrated in Fig. 1.2, our contributions include two parts:

Firstly, we study the least square estimation problem in wireless sensor networks, i.e. the measurements from sensors main contain noise, thus redundant measurements that are used to reduce the effects of error are often sampled to form an over-determined system. The primary objective of the wireless sensor network is to estimate the parameters of interest based on the measurements from all sensors in the network. Due to the very limited amount of energy and computation power available on sensor nodes, it is imperative to design new algorithms to perform least squares estimation in a distributed fashion. Recent years, researchers have proposed a wide range of strategies for distributed least squares estimation. However, each strategy has its own design objectives and applications scenarios. No uniform scheme exists for current practical usage, making it difficult to evaluate their relative effectiveness and performance. Thus, we propose a 3 dimensional framework which provides a basis for designing, analyzing and evaluating strategies to address parameters estimation issues using least squares estimation algorithms in wireless sensor networks. In the 3D framework, we propose three design aspects of designing distributed least squares estimation, and then we study the existing works from the design aspects and then discuss their advantages and disadvantages correspondingly.

Then, based on our proposed framework, we focus on design new distributed estimation algorithm for least squares while we wish to conserve energy by minimizing communication with our design, constraints on communication delays will also need to be satisfied. Thus, we propose E^3, a new tree-based distributed algorithm specifically designed to guarantee the precision of least squares estimation in sensor networks, with the objective of minimizing the energy consumption incurred during communication, while observing constraints on application-specific communication delays. To realistically evaluate the performance, we conduct structural damage detection experiments in a real SHM platform to test our algorithms. Compared to previous works, we show that E^3 maintains the same level of estimation precision while incurring much lower energy costs. At last, we address that our 3D framework can facilitate the design, classification and evaluation of the current distributed least squares estimation strategies in sensor networks.

These two parts can be integrated for distributed least squares estimation algorithms design. The first part is a general framework for distributed least squares estimation algorithms design.Our proposed 3D framework contains the existing distributed least squares algorithms which considered three design aspects: aggregation structure, performance metric and communication pattern. The second part is E^3 which is a specific design for applications with delay constraints and energy consumption optimization requirements. It can achieve a balance between the estimation delay and total energy consumption in the entire network. E^3 is one of the specific strategies which can guarantee the precision of least squares estimation in sensor networks, with the objective of minimizing the energy consumption incurred during communication, while observing constraints on application-specific communication delays.

1.6 Organization of the Thesis

The structure of this thesis is shown in Fig. 1.2. Chapter 1 is the introduction to this thesis. Chapter 2 reviews related works in the literature. The main body of this thesis is

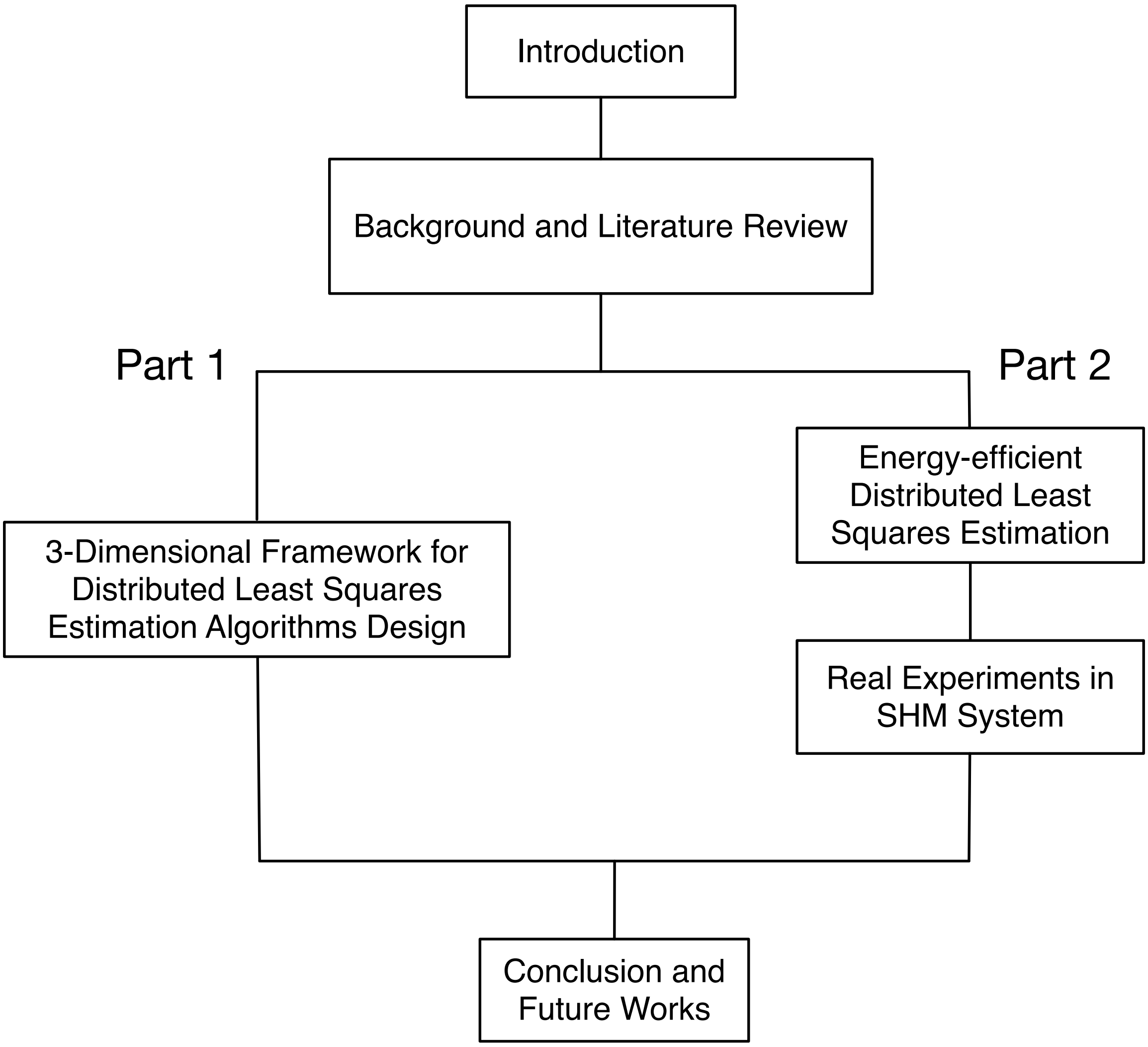


Fig. 1.2: An outline of the contributions in this thesis

divided into two parts from Chapter 3 to Chapter 4. The details are presented as follows.

In the first part, we mainly discuss the general framework of distributed least squares estimation algorithm design from three design aspects in Chapter 3. They are computation architecture, optimized performance and communication pattern. In this part, we investigated the existing distributed algorithms, fit them into our 3D framework and then discussed their corresponding advantages and disadvantages.

In the second part, we make full use of our proposed framework to design new strategies to perform least squares estimation in distributed manner in Chapter 4. We mainly discuss our specific design, E^3 for natural frequency estimation of the civil engineering structures. To realistically evaluate the performance, structural damage detection experiments in a real SHM platform are conducted to test our algorithms.

Finally, we conclude the thesis and discuss the directions of future works in Chapter 5.

Chapter 2

Literature Review

In this chapter, we review existing works about signal-processing algorithms in the applications of WSNs. As we have discussed, we focus on distributed algorithms design for computation-intensive applications in this thesis. In Section 2.1, we review the existing works about applications with data-intensive and computation-intensive features in WSNs. Then we review the existing works about distributed least squares estimation algorithms in WSNs in Section 2.2.

2.1 Existing Works on signal-processing algorithms in the applications of WSNs

According to the time complexity and the level of collaborations among different sensors, typical signal processing algorithms (SPA) in WSNs can be categorized as (1) lightweight (2) computationally intensive collaborations of multiple sensors.

2.1.1 Computation Lightweight SPA

Most of the signal-processing algorithms in the applications of WSNs belong to the first two categories. Algorithms in the first category usually include those whose objective is to find some statistics (e.g. mean, maximum, minimum) of measurement data of sensor nodes over some region and/or over a period of time,such as monitoring fire in a forest, temperature in a building, etc. [MCP$^+$02] [KPKK07] [YWM05]. These applications are characterized as having low data rates and with light-weight computations. Designing the corresponding distributed version is straightforward. For example, each sensor only needs to collect data from its children and implements some certain aggregation function (e.g. mean, maximum, minimum). These studies were focused on how to establish the best routing in the network.

2.1.2 Computation Intensive SPA

Computation intensive with feature-level or decision level collaborations SPA in the second category are shown in Fig 2.1(a) and Fig. 2.1(b). Feature-level or decision-level collaboration allows data from each sensor to be processed first and then the local results, which uses much fewer bits than the original one, are combined together. Particularly, in Fig. 2.1(a), data from each sensor is processed individually to obtain a local decision, and then all the decisions are combined together through decision-fusion. In Fig. 2.1(b), each sensor extracts a feature and all the features are combined through feature-fusion techniques. For these algorithms, designing the distributed version is relatively simple: each sensor can process its own data independently without exchanging information with others, and then the local results (either local features or local decisions) are transmitted to a sensor for fusion afterwards. The studies are usually focused on how to decrease the computation at each sensor, usually via replacing floating-point operations with integer arithmetic operations without operations or assigning computationally-intensive operations to the base station.

However, a large amount of algorithms in WSNs belong to data-level collaborations SPA: they are computationally intensive and with data-level collaboration of multiple sensors. Data-level collaboration prohibits the probability that each sensor can process its own data independently without exchanging information with others. In addition, in a typical algorithm with data-level collaboration of multiple sensors, data from different sensors are tightly coupled in the computation task and cannot be easily decomposed into smaller subtasks. To design a distributed version for the algorithms with architecture shown in Fig. 2.1(c), one commonly adopted approach is to design the computation components as shown in Fig. 2.1(a) or Fig. 2.1(b) (e.g. feature extraction/fusion, decision making/fusion) such

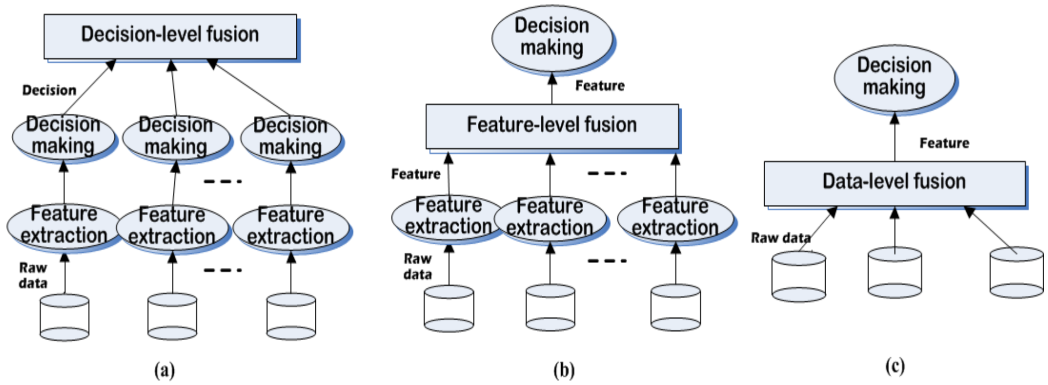


Fig. 2.1: Algorithms with different levels of collaboration(a)at decision level(b)at feature level(c)at data level

that the Fig. 2.1(c) can be implemented as in Fig. 2.1(a) or 2.1(b). For example in SHM, to identify structural vibration characteristics called modal parameters in a distributed way, a cluster-based approach is proposed in [Abe90] (see Fig.2.1). This approach tries to map Fig.2.1(c) to Fig.2.1(b): the whole network is divided into a number of clusters. Each cluster has a cluster head which is responsible for obtaining the local modal parameters for that cluster, and then all the local modal parameters are stitched together to obtain the global result. Similarly, the distributed signal detection and data fusion in multi-sensor systems [XV96] [KZG92] [T$^+$93] also follow this approach. However, existing mechanisms to design distributed versions of computation intensive signal processing algorithms have the following disadvantages:

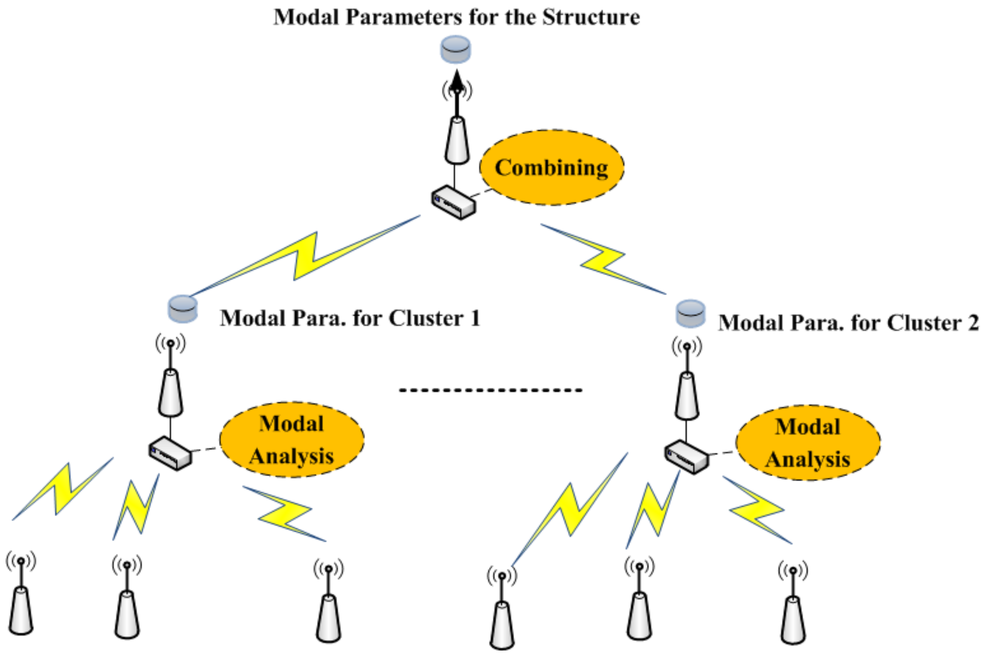


Fig. 2.2: Cluster-based modal parameter estimation

1. Resource usage: When designing a distributed algorithm, the problem of how to optimally utilize the available resources (energy supply, wireless communication bandwidth, computational capability) has not been fully addressed.

2. Quality: Existing works on designing distributed algorithms usually cannot guarantee that the results of the distributed version have the same accuracy as the centralized one. For example, in the clustering approach [Abe90], it should be noticed that the accuracy of the 'stitched results' cannot be guaranteed to be comparable with the original centralized one.

At last, it should be noticed that the problem to design a distributed version for the data-intensive and computation-intensive applications in WSNs is different from what has been studied extensively in parallel and distributed computing [Zom96] in the following aspects:

1. Objective: Parallel and distributed computing generally focuses on how to utilize the computation power of a given number of computation utilities such that the time-to-completion of a computation task can be minimized. The computing utilities generally do not generate data. On the contrary, each sensor node in the network is not only the computation entity, but also the resource of the data. Considering the limited energy, energy cost (including communication and computation cost) is generally more important than time-to-completion. Accordingly, when we decompose a computation task into smaller ones, it is highly desirable that a computation entity, when implement its sub-task, use only its own data or data from its one-hop neighbors.

2. Computation constraints: Various parallel algorithms have also been developed to speed up the execution of these methods . However, designed for high-performance computers, these approaches need significant amount of computational/memory resources and require the knowledge of global information. As a result, they cannot be executed by a distributed system.

2.2 Existing Works on distributed least squares estimation algorithms in WSNs

Sensor networks have found widespread adoption in many domain-specific applications, such as structural health monitoring (SHM) [Avi01], volcano monitoring [KSS13], etc; most of which have a data-intensive nature. For example, to detect possible structural damage, each accelerator deployed in a structure needs to sample in the range of 16-24 bit at 200-1000Hz [DFPS96]. It is typical that least squares estimation, which is used for estimating some parameters of interest from a large number of redundant measurements, serves as a critical component in these applications' algorithms. Due to the limited capability and resources of wireless sensor networks and domain specific characteristics, distributed algorithms for least squares estimation in terms of energy and real-time implementation are required.

There exist some works over adaptive networks in the literature, including incremental based strategies [CS11] [LS06] [SL07] [SL06] and diffusion based strategies [BMS11] [LS07] [LS08] [CLS08], focusing on the estimation performance. The incremental recursive least squares algorithm (I-RLS)[SL06] is one of the first such schemes, which sequentially aggregate new raw measurements with intermediate results while performing least squares estimation. The I-RLS can achieve the exact global solution of the least squares system. However, a Hamilton path across sensors for implementing I-RLS which has been proved a NP-hard problem is required, and the challenges still remain in large scale wireless sensor networks. Besides, it also limits the practical usage in real-time applications using large-size sensor networks because of the long delay incurred by long transmission path.

Another category is based on diffusion strategy. One called diffusion recursive least squares, which is also based on the recursive scheme, was proposed in[CLS08]. Diffusion based schemes allow global results to be spread over the entire network. Nevertheless, in addition to local estimates, each node is required to diffuse its raw measurements to its one-hop neighbors. Continuous local communication and the presence of the communication noise incur estimation performance degradation. Only after certain times of iterations can the final results converge to the global ones. As a consequence, the diffusion methods are significantly limited in practical use of certain applications in sensor networks in terms of energy and real-time implementation.

Several distributed estimation algorithms, which are rooted on iterative optimization methods, have also been investigated. The distributed multisplitting method is one of such strategies [Ren98]. It is based on stationary iterative methods in which each sensor does the same routines based on its own measurements. However, each sensor is required to flood its information to the entire network for cooperation in each iterative round. Continuous flooding in large-size sensor networks is not practical in terms of energy efficiency. Such limitation is also shared by other methods in this category.

Chapter 3

3D Framework for Distributed Least Squares Estimation Algorithms Design for Sensor Networks

In this chapter, we investigate the distributed least squares estimation algorithms in sensor networks. We propose a 3D framework which is able to provide a basis for designing, analyzing and evaluating strategies to address parameters estimation issues using least squares estimation algorithms in wireless sensor networks. This chapter is organized as follows: Section 3.1 is the overview of this work. Section 3.2 describes the problem and the network model. Section 3.3 describes the main design aspects for distributed least squares estimation in sensor networks. The 3D framework is proposed in Section 3.4. Finally, Section 3.5 concludes this chapter.

3.1 Overview

The trend toward the real-deployment of wireless sensor networks in numerous application domains including intelligent transportation system, structural health monitoring (SHM) of large buildings and bridges, states-estimation in smart grid, etc., is increasingly demanding for efficient signal processing in wireless sensor networks. However, due to the very limited energy and computation capabilities in sensor networks, traditional centralized signal processing strategies by no means straightforward to implement. Moreover, the traditional centralized fashion, where sensor nodes forward their measurements to a central server and they are processed to estimate the parameters of interest, results in long delay especially in large scale wireless sensor networks. These challenges make the traditional centralized signal processing algorithms are unsuitable in wireless sensor networks, emphasizing the need for efficient signal processing strategies.

Fortunately, many domain-specific problems can usually be formulated as a large linear least squares system based on measurements from sensors. These measurements may contain noise, thus redundant measurements that are used to reduce the effects of error are often sampled to form an over-determined system. As a result, in many domain-specific applications, efficient signal processing strategy design is narrowed down to design and implement efficient algorithms to solve such a large linear least squares problem and obtain the optimal global estimates, where each sensor node has only the partial independent rows of the least squares system.

An intuitive, and perhaps naive way, is to solve the problem in a centralized fashion: sensor nodes forward their measurements to a central server, where they are processed to estimate the parameters of interest. Even if we do not consider the computation power that may be needed at the central server, such a centralized solution is still not practical in sensor networks, due to the significant amount of communication cost and the long delays that are required to deliver the data to the server. A distributed solution that processes the data within the large sensor network itself is, therefore, a much more favourable choice.

By a distributed solution, it is implied that the least squares computation will be performed in the sensor nodes themselves, and that only the result of such computation and part of measurements will be transmitted. Though such in-network processing may conserve a significant amount of energy due to the reduced amount of data to be transmitted, the delay of finishing the computation may be longer, and the accuracy of the results may be sacrificed. Recent years, researchers have proposed a wide range of strategies for distributed least squares estimation. However, each strategy has its own design objectives and applications scenarios. No guided scheme exists for current practical usage, making it difficult to evaluate their relative effectiveness and performance.

In this chapter, we focus on the problem of distributed least squares estimation algorithms design in large-scale sensor networks, which is an important foundation in many domain-specific sensor network applications. With distributed computation, we stipulate that each sensor node will process information locally with observations from itself and some of its immediate neighbours. In this context, we develop a general 3 dimensional model that captures the main features of distributed least squares estimation schemes and provides a basis for evaluating existing strategies. As far as we know, our framework is the first one to demonstrate the distributed least squares estimation within large wireless sensor networks with guaranteed accuracy and other optimized performance(energy, delay, etc.) in a general manner.

3.2 System Model

Consider a wireless sensor network with N sensor nodes (or *nodes*), modeled as an undirected graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges representing the communication links between pairs of nodes. Though each node is only able to communicate with its immediate neighbors directly, messages may be relayed via multiple hops to a destination node. Without loss of generality, we assume that the diameter of the network is $\log N$ (i.e., any message can be sent from one node to another through at most $\log N$ hops). Also, let's assume that each link in the network between neighboring nodes has unit bandwidth and each node only has one radio. Therefore, the communication delay of one unit data delivery between direct neighbors (either through a unicast to one direct neighbor or multicast/broadcast to all direct neighbors) would be one unit time. Notice that, here we assume the link layer supports broadcast which is often true in many sensor networks. For simplicity, according to the description in [SSK$^+$], we also use the term *broadcast* for local broadcast to one-hop neighbors and *flood* for network flooding.

In such a sensor network, the primary objective is to collectively estimate M unknown parameters which form a $M \times 1$ column vector, denoted by ω^0, using the least-squares estimation method. Each node $k \in \{1, \ldots, N\}$ have access to one pair of measurements: scalar measurements $d_k \in \mathbb{R}^{(t_{\text{rows}} \times 1)}$ and a regression matrix $u_k \in \mathbb{R}^{(t_{\text{rows}} \times M)}$, where t_{rows} denotes the number of rows that each matrix u_k contains.

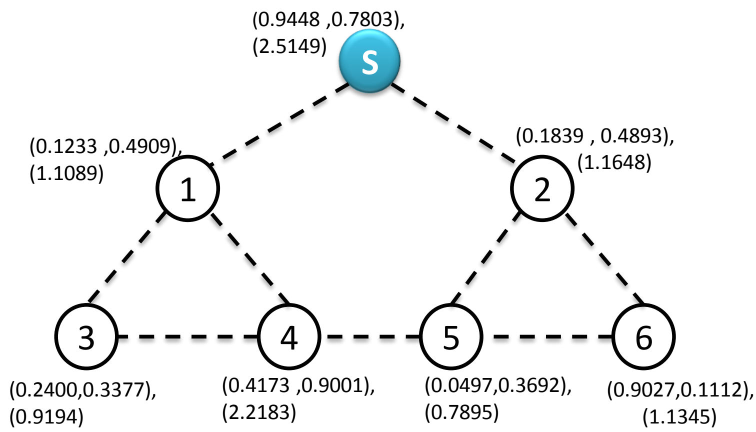


Fig. 3.1: A wireless sensor network: an example of our system model.

Let us consider an example, shown in Fig. 3.1. The dashed lines denote the undirected edges between nodes. The network contains 7 nodes and one of them is the sink node. According to our system model, we can see that the measurements in sensor 1 in our example are $u_1 = (0.1233, 0.4909)$ and $d_1 = 1.1089$, where $M = 2$ and $t_{\text{rows}} = 1$. The vector y corresponds to all the measurements from all of the nodes, while the matrix H

corresponds to all the regression data. Then we can have

$$y = \text{col}\{d_1, d_2, d_3, \ldots, d_N\}(t_{\text{rows}} N \times 1) \tag{3.1}$$

$$H = \text{col}\{u_1, u_2, u_3, \ldots, u_N\}(t_{\text{rows}} N \times M) \tag{3.2}$$

For a system with N nodes, the equation of the entire system would be

$$H \times \omega^0 = y \tag{3.3}$$

In our example the least squares system constructed at the sink node, based on 7 pairs of sensing data, is shown in Eq. (3.4):

$$\begin{pmatrix} 0.1233 & 0.4909 \\ 0.1839 & 0.4893 \\ 0.2400 & 0.3377 \\ 0.4173 & 0.9001 \\ 0.0497 & 0.3692 \\ 0.9027 & 0.1112 \\ 0.9448 & 0.7803 \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1.1089 \\ 1.1648 \\ 0.9194 \\ 2.2183 \\ 0.7895 \\ 1.1345 \\ 2.5149 \end{pmatrix} \tag{3.4}$$

Without loss of the generality, our network model accounts explicitly for non-ideal sensor to sensor links, which implies that each sensor's measurements are corrupted by a noise ν_k, and ν_k is assumed as a zero-mean temporally and spatially uncorrelated Gaussian white noise process with variance σ_{ν_k}. Then we can have

$$y = H \times \omega^0 + \nu \tag{3.5}$$

Typically, the centralized and non-iterative least squares estimate of ω^0 of Eq. (3.5) can be calculated as

$$\omega^0 = (H^T H)^{-1} H^T y \tag{3.6}$$

In our running example in Fig. 3.1, the noise variance of each node $\sigma_{\nu_k} = 0.01^2$. The objective of a centralized computation is to collect the 7 pairs of measurement data from the sensor nodes to the sink node, where the two unknown parameters (x, y) will be estimated. With the traditional least squares estimation method, Eq. (3.4) is solved by using Eq. (3.6) directly.

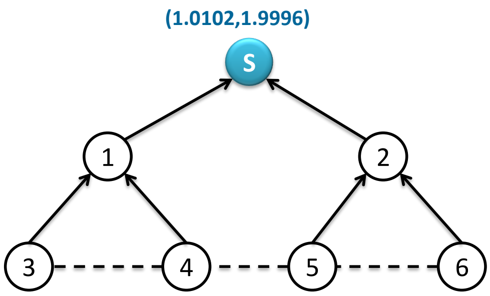


Fig. 3.2: Computing the least squares estimation at the sink node in a centralized manner, while measurement data is being transmitted in a shortest path tree to minimize the transmission cost.

In order to minimize the data transmission cost when measurement data is being transmitted from the sensor nodes to the sink, we use the shortest path tree rooted at the sink as the routing strategy, as shown in Fig. 3.2. When we consider the calculation for the total amount of data to be transmitted, we regard a scalar value in a matrix transmitted over one hop as a unit. In our example, the data size of measurements over one hop of transmission at each sensor is 3. There are four nodes whose measurements need to be transmitted over two hops to the sink node. Thus, the amount of data transmitted in total is $4 \times 2 \times 3 + 2 \times 1 \times 3 = 30$. After the transmission completes, the sink node estimates the parameters by using Eq. (3.6), with the result being $(x, y) = (1.0102, 1.9996)$. The estimation error is 3.01% by calculating the square root of the sum of the mean square. If, in addition to the transmission cost, we consider the maximum number of hops (*hop count*) from any sensor node to the sink, it is 2 in our example.

Even from our simple example, we can clearly see that the routing strategy that we used to transmit measurement data to the sink leads to a high transmission cost. In addition, by solving Eq. (3.4), complicated matrix operations such as matrix inversion and matrix transposition are needed, which is expensive to be performed in a centralized manner in a large scale sensor network. A distributed algorithm is naturally preferred.

Due to the data-intensive nature of the domain-specific applications, the formulated least square systems are always quasi-over-determined. Therefore, we need to add some form of regularization to avoid strong, undesired influence of small singular values dominating the solutions. This can be achieved by applying a regularization parameter for determining the least-square solutions. Then the least squares estimation problem can be formulated as follows according to [CLS08]:

$$\omega = \arg\min_{\omega}\{\lambda^{i+1}\|\omega\|^2_{\Pi} + \|y - H \times \omega\|^2_w\} \tag{3.7}$$

where $\lambda \in (0, 1]$ is a forgetting factor, $\prod > 0$ is a regularization matrix and usually $\prod = \delta^{-1} \cdot I_M$ $(\delta > 0)$ is large, and $w \geq 0$ is a weighting matrix which is related to the sensing noise of each node. We do not focus on the optimization of regularizing parameters in this chapter.

3.3 Design Aspects of Our 3 Dimensional Framework

The trend toward the real-deployment of sensor networks in more application domains drives us to propose a general framework for distributed algorithms design. In this section and Sec. 3.4, we propose a general 3 dimensional framework as a basis for designing, analyzing and evaluating distributed estimation algorithms including but not limited to LSE in WSNs. In this chapter, we use least squares estimation as a case for our general framework. Essentially, a distributed least squares estimation strategy to solve ω^0 in Eq. (3.5) in sensor networks can focus on three design aspects: *aggregation structure*, *performance metric* and *communication pattern*.

3.3.1 Aggregation Structure

A distributed algorithm implies to utilize the computation capability of each sensor. In this sense, we wish to carry out the computation on the route when each node sends its data to the sink(Hamiltonian path based algorithm [SL06]) or each sensor has the final result once the estimation algorithm converged(e.g. Diffusion-based algorithm [CLS08]). This indicates that each sensor should carry out certain computation task using its own or received data and communicate local results with its neighbors. To achieve this goal, the aggregation structure should be determined. Then, based on the aggregation structure and the graph of the deployed sensors, the optimal routing can be obtained.

We have investigated general aggregation structures for typical sophisticated signal processing algorithms including but not limited to LSE. The structures should allow both computation and associated data are distributed within the network. Accordingly, we can have a number of aggregation structures shown in Fig. 3.3. For example, Fig. 3.3(a) shows a cluster-based scheme in which cluster heads collect and process the data from the corresponding cluster members. The local results from different clusters are then combined to obtain a global result. Besides clustering, another aggregation structure can be a Hamiltonian path shown in Fig. 3.3(b). In this structure, the results are updated along the path and when the results reach one en-route node, they will be updated by the node's data. Similarly, Fig. 3.3(c) shows the possibility to update along a backbone path. A backbone

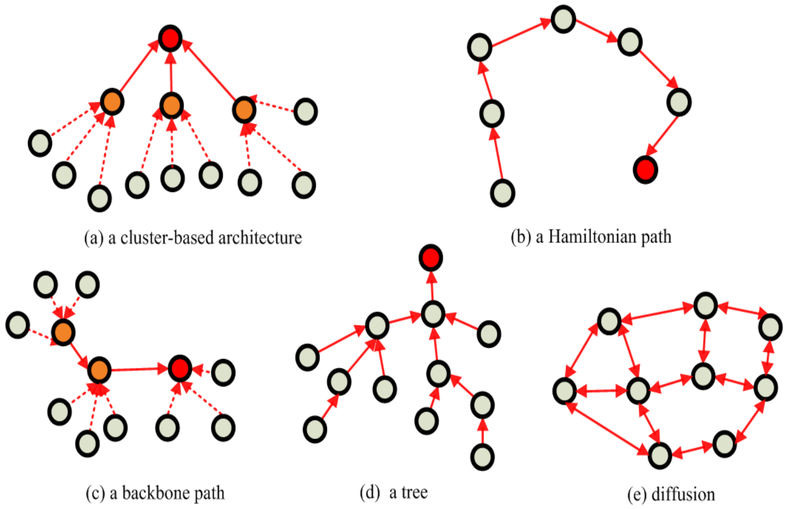


Fig. 3.3: Different aggregation structures for sophisticated algorithms

path is a path that all the sensor nodes are either on the path or have neighbors on the path. The nodes on the path are the backbone nodes and the rest the leaf nodes. Using this structure, leaf nodes transmit their raw data to their corresponding backbone nodes. The backbone nodes process the received data from both leaf nodes and the preceding backbone node(if there is any) and transmit the result to the next backbone node. In this way, the result is updated when it travels along the backbone path. The fifth possible network structure is tree shown in Fig. 3.3(d). In this structure, each node in the tree collects data (either in the form of raw data or intermediate results) from its children, processes it and sends the result to its parents. Fig. 3.3(e) shows a diffusion-based aggregation structure in which each node communicates with all or part of its neighbors. At every instant, the local result is combined with information from the neighboring nodes in order to improve the result at the local node. In steady-state, after sufficient observations and cooperation, the nodes would eventually converge to the global results. For clarifity, we classify the aggregation structures into four categories: path-based(P) Fig. 3.3(b), tree-based(T) 3.3

(c)(d), cluster-based (C) Fig. 3.3(a) and diffusion-based(D) Fig. 3.3(e).

3.3.2 Performance Metric

The fundamental objective of sensor networks for the domain-specific applications is to estimate the parameters of interest. To achieve this objective, three performance metrics need to be considered: accuracy(A), delay(De) and energy consumption(Ec). Firmly, the estimation accuracy is the most fundamental requirement for a distributed algorithm design. Ideally, we hope that the distributed strategy can achieve the same accuracy with the centralized approach. Besides, the time taken for the network to finish an estimation is referred to as delay, which shall include the consideration of the message size and number of hops the data packet traversed. In this paper, we focus on the communication delays while ignoring the computation time in each node. With the Moores law, the computation capability is increasing faster than the communication capacity of transceivers. The communication delay is typically dominating the computation time. Finally, to solve a least squares problem of large size, the communication cost is one of the most influential performance metrics. Here, we refer the energy consumption as the cost involved in the messages exchanged in the networks during an estimation. For simplicity, we use the total data transmitted in the entire network for an estimation representing the energy consumption.

To the best of our knowledge, there is no existing general distributed algorithm which can optimize the three performance metrics simultaneously, while existing algorithms are designed for certain specific applications. Practically, to design a practical strategy for solving least squares estimation in sensor networks requires sacrificing some of the performances(e.g. delay) to reach other optimization objectives(e.g. energy consumption). For clarity, we list the performance priority level for each algorithm. As discussion, accuracy is the basic performance metric needed to be guaranteed. When we refer to AEc strategy, we mean accuracy guaranteed and energy consumption minimization distributed algorithms, while ADe is accuracy guaranteed and delay minimization strategy. For simplicity, some existing algorithms which have strict accuracy requirements while other performance metrics(De and Ec) are not explicitly considered are classified into the accuracy focused (AC) category.

3.3.3 Communication Pattern

Performing least squares estimation in distributed fashion implies that sensor nodes will process information locally with observations from itself and some of its immediate neighbours. Thus, communication among sensor nodes is required. In this paper, we assume the link layer supports broadcast in network. For simplicity, we consider a one-hop broadcast as multiple unicasts. Then we can classify the communication pattern in the sensor networks into three categories: unicast(one-hop unicast or multi-hop unicast), one-hop broadcast (local broadcast to one-hop neighbor nodes) and network flooding (broadcast the message to all the nodes in the entire network). Actually, most existing distributed least squares estimation algorithms use not only one single communication pattern. They usually are unicast and one-hop broadcast, unicast and flood, .etc. However, each strategy has one dominate communication pattern which contributes most to the entire network communication cost. For simplicity and convenience, in the rest of the paper we use the term unicast(U) for strategies for which unicast is the dominating communication pattern. Likewise, we use the term broadcast(B) and flood(F) in the rest of the paper.

3.4 3D Design Framework

Figure.3.4 shows a 3D framework that represents each design aspect as one orthogonal dimension. Because the three dimensions are independent of each another, a wide variety of distributed least squares estimation strategies can be created by combining properties from each.

A XX-YY-ZZ string expresses a strategy in which XX represents the aggregation structure, i.e., P(path-based), T(tree-based), C(cluster-based), D(diffusion-based), .etc. YY stands for performance metrics, i.e., ADe, AEc, AC.etc. , and ZZ symbolizes the dominating communication pattern which contributes most to the communication cost in the entire network i.e., U(Unicast), F(Flood), B(Broadcast), .etc. A strategy's overall configuration has a special value for each parameter. Some XX-YY-ZZ combinations are meaningful in the domain-specific applications using wireless sensor networks. In fact, all the existing distributed least squares estimation schemes in the literature have corresponding XX-YY-ZZ tuples.

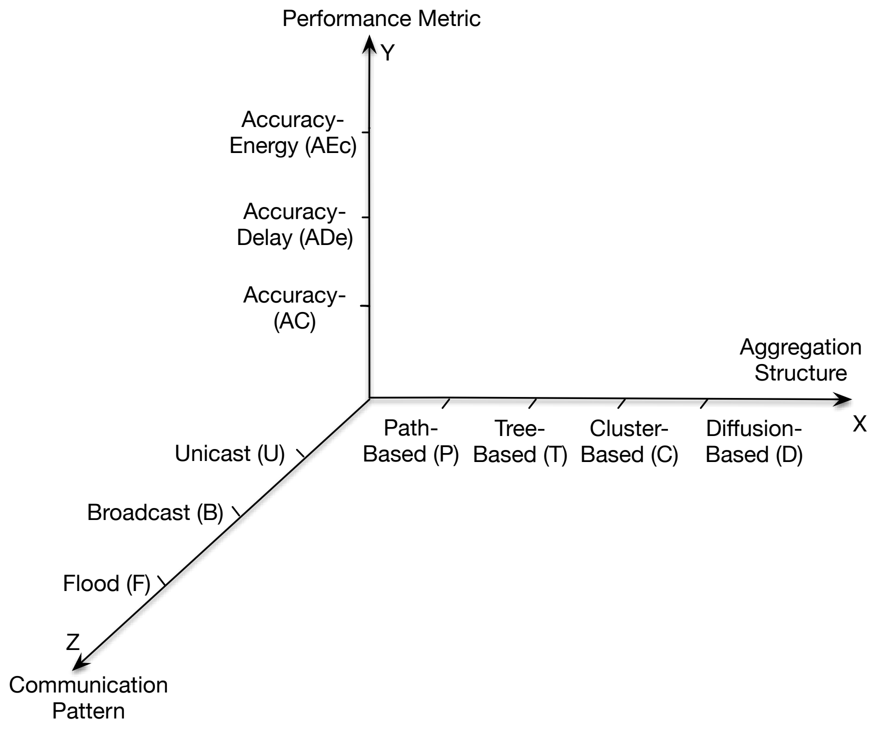


Fig. 3.4: 3D design framework. Each axis represents one design aspect and contains a range of properties.

P-AEc-U is identical to the distributed recursive least squares estimation(DR-LSE) strategy[SL06] in sensor networks. DR-LSE is a scheme which sequentially aggregate new raw measurements with intermediate results along a Hamiltonian path which traverses all of the sensor nodes in the entire network. The computation pattern is unicast. Besides, DR-LSE can achieve the exact global solution of the least squares system which means that the estimation accuracy can be guaranteed. The results converge once all of the nodes are involved in the estimation process. Furthermore, each sensor aggregates the intermediate results with its raw measurements and transmits the updated results which data amount is a constant depended on the scale of the network. DR-LS has been shown to outperform most other existing works in terms of estimation accuracy and energy efficiency. However, one problem is that building a Hamiltonian path has been proved to be a NP-hard problem, from which the path does not always exist can be down once one sensor is down.

Diffusion distributed least squares estimation[CLS08] is a kind of D-AC-B strategy that it is a diffusion-based manner which has no topology constraint and mainly focuses on accuracy. The estimation process is continuous while each sensor is sensing the measurements till all of the local estimates have converged to a pre-defined accuracy level. Estimation delay is mainly determined by the converged speed of the estimation algorithm. Diffusion based schemes allow global results to be spread over the entire network, i.e., there is no requirement for a sink node to collect and combine the intermediate results. However, each node is required to communicate with its one hop neighbors continuously in which we called the computation pattern one-hop broadcast. The advantage is that there is no topology constraint for dense sensor networks. However, for sparse networks, local raw measurements from itself and its neighbours' are not enough to make the local sub-least squares system solvable. It means that, the final results in each node may not able to converge. Besides, continuous local communication and the presence of the communication noise incur estimation performance degradation.

Divide and conquer approach breaks down the global least squares system into two or more sub-least square(the raw measurements overlap can exist) problems, the solutions to the sub-problems are then combined to give a solution to the original problem[Abe90]. In [Abe90], the authors only considered the signal processing part, i.e., data division and intermediate results combination. We slightly modified their approach to be suitable in sensor networks. Assume that each node contains one row of the coefficient matrix. Based on the original algorithm, the divide and conquer approach to least square estimation process in sensor network contains two steps. First, the entire network is divided into clusters, and then clustering heads collect all of the row measurements from its cluster members so as to solve a sub least-squares problem, and second, the clustering head transmits its local estimates to sink node which all of the sub solutions are combined. The communication pattern of this strategy is unicast (either can be one-hop unicast or multi-hop unicast). The inconvenience of this approach is that how to divide the entire network into clusters to achieve the minimum transmission cost and guarantee the accuracy of the final results simultaneously. Obviously, this approach is one kind of cluster-based schemes which belongs to C-*-B(in which * represents those situations with nonfixed attributes and can be any combination of properties along the dimension).

Recently a survey paper [SSK^{+}] has analysed and discussed one category of distributed strategies–distributed least-squares iterative methods in mesh networks as shown in Fig. 3.4. We list them in the following. Since all of the distributed iterative methods shown below are designed based on the existing centralized iterative methods, the estimated results can be guaranteed especially in ideal network. Besides, they required global information when calculating some intermediate parameters. For example, D-MS. Thus, network flooding is necessary when performing distributed iterative methods in sensor networks, which requires each node to exchange information continuously. As a consequence, this category introduces large communication costs and a high collision probability due to the large number of nodes transmitting simultaneously. The advantage is that there is no limitation in the network topology. However, the estimation delay is mainly determined by the convergence speed of the computation algorithm itself and is without topology limitation. This category of distributed strategies belongs to D-AC-F.

- D-MS Distributed Multisplitting method
- D-MCGLS Distributed Modified Conjugate Gradient Least-Squares method
- D-CARP Distributed Component-Average Row Projection method
- D-CE Distributed Cooperative Estimation methods
- D-LMS Distributed Least Mean Squares method

3.5 Summary

In this chapter, we have proposed a 3 dimensional framework for distributed least squares estimation algorithms design in sensor networks. Our framework shows that distributed LSE algorithms design mainly focus on three design aspects: *aggregation structure*, *performance metric* and *communication pattern*. Based on our proposed framework, we have investigated existing works on distributed LSE and have analysed their advantages and disadvantages, respectively.

Chapter 4

E^3: Towards Energy-Efficient Distributed Least Squares Estimation in Sensor Networks

In this chapter, we investigate the characteristics of signal processing algorithms in civil engineering domain. After that, based on our proposed framework in Chapter 3, we propose a specific design of distributed algorithm, E^3, for structural damage detection, from which it combines the advantages of T-AEc-U and T-ADe-U strategies in sensor networks. This chapter is organized as follows: Section 4.1 is the overview of this work. Section 4.2 describes the system model and the formal problem formulation. Following this is the distributed algorithm design for solving this problem in Section 4.3. Section 4.4 reports the simulation results and experiments results, respectively, and finally Section 4.5 concludes this chapter.

4.1 Overview

Due to low-cost and ease of deployment, WSNs are emerging as sensing paradigms that the structural engineering domain has begun to consider as substitutes for traditional wired SHM systems. The objective of SHM is to monitor the integrity of structures and detect and pinpoint the locations of any possible damage. In a typical *wire-based* SHM system as shown in Fig. 1.1, an array of sensors, usually accelerometers, are deployed on different locations of a structure. These nodes collect the structure's responses in a synchronous manner and transmit them through cables to a central server where one or more SHM algorithms are implemented to extract damage-sensitive vibration characteristics. By examining these characteristics, damage can be detected and located [DFPS96].

Using WSNs as alternatives to traditional wired SHM systems can reduce the cost and deployment time but introduces many challenges. These challenges are mainly due to two properties of SHM applications. Firstly, the high sampling frequency of SHM applications ($>$ hundreds of Hz) generates a large amount of data which can reach thousands or even tens of thousands for each sensor in a single round of data collection process. Thus the SHM applications becomes quite data intensive. Secondly, damage detection usually involves intensive computations such as large matrix operations. Consequently, distributed processing, by which only important information, rather than all raw data, needs to be transmitted, is highly preferable.

Fortunately, least squares estimation is a foundation for one of damage detection algorithms originally proposed by civil engineers [Gol], from which damage information can be analyzed. By a distributed solution, it is implied that the least squares computation will be performed in the sensor nodes themselves, and that only the result of such computation will be transmitted. Though such in-network processing may conserve a significant amount of energy due to the reduced amount of data to be transmitted, the delay of finishing the computation may be longer, and the accuracy of the results may be sacrificed. It is non-trivial to achieve a similar level of accuracy as the centralized solution, or to work with constraints in the delays of computing global estimates.

Thus, in this chapter, we focus on the problem of distributed least squares estimation in sensor networks which can be practically used in SHM applications. With distributed computation, we stipulate that each sensor node will process information locally, with observations from itself and some of its immediate neighbors. In this context, we focus on *accuracy*, *energy consumption*, and *delay* for completing the computation. Based on our proposed framework for distributed LSE algorithms design in Chapter 3, we propose a tree-based strategy which can combine the advantages of T-AEc-U and T-ADe-U strategies. Inspired by [Abe90], we use fisher information as a gauge of *accuracy*, with which a quantity can be estimated from partial observations. Our objective in this chapter is to arbitrate the conflicts between energy consumption and delays, without sacrificing the accuracy of least squares estimation.

To achieve this objective, the highlight of our original contribution is E^3, a new distributed algorithm which can guarantee the accuracy of least squares estimation in sensor networks, with the objective of *minimizing the total energy consumption* when solving the over-determined system, subject to a *delay constraint*. The energy consumption is measured by the communication cost of the entire network, while the delay is measured by the longest hop count from the leaf to the sink node during the distributed computation process. The outcome of our proposed distributed algorithm is the routing strategy for the distributed estimation process. To validate the efficacy and effectiveness of our algorithm, we show results from both simulations and a real-world test bed, with data from an actual structural health monitoring system deployed in a building.

4.2 System Model and Problem Formulation

We first describe the system model used in this work. Then we formulate the problem.

4.2.1 System Model

Consider a wireless sensor network with N sensor nodes (or *nodes*), modeled as an undirected graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges representing the communication links between pairs of nodes. Though each node is only able to communicate with its immediate neighbors directly, messages may be relayed via multiple hops to a destination node.

In such a sensor network, our primary objective is to collectively estimate M unknown parameters which form a $M \times 1$ column vector, denoted by ω^0, using the least-squares estimation method. Each node $k \in \{1, \ldots, N\}$ has access to one pair of measurements: scalar measurements $d_k \in \mathbb{R}^{(t_{\text{rows}} \times 1)}$ and a regression matrix $u_k \in \mathbb{R}^{(t_{\text{rows}} \times M)}$, where t_{rows} denotes the number of rows that each matrix u_k contains.

Due to the data-intensive nature of the domain-specific applications, the formulated least square systems are always Quasi-over-determined. Therefore, we need to add some form of regularization to avoid strong, undesired influence of small singular values dominating the solutions. This can be achieved by applying a regularization parameter for determining the least-square solutions. Then the least squares estimation problem can be formulated as follows according to [CLS08]:

$$\omega = \arg\min_{\omega} \{\lambda^{i+1} \|\omega\|_{\Pi}^2 + \|y - H \times \omega\|_w^2\} \tag{4.1}$$

where $\lambda \in (0, 1]$ is a forgetting factor, $\prod > 0$ is a regularization matrix and usually $\prod = \delta^{-1} \cdot I_M$ ($\delta > 0$) is large, and $w \geq 0$ is a weighting matrix which is related to the sensing noise of each node. We do not focus on the optimization of regularizing parameters in this chapter.

4.2.2 Problem Formulation

With each node performing its local computation based upon the information exchanged with its immediate neighbors, we wish to achieve the accuracy of a centralized least squares estimator, as defined in our system model, yet with a distributed algorithm. By performing local computation on each node and relaying partial computation results to the sink node, our distributed algorithm should attempt to compute an optimal routing strategy to minimize the energy consumed in the entire sensor network to complete the computation, subject to a delay constraint. The delay is measured by the largest hop count from leaf nodes to the sink node in the distributed computation process, and the energy consumption is measured by the total amount of data transmitted during the process.

The essence of our new algorithm is to perform estimation while collecting data within the sensor network. When the data arrives at the sink, the same approximation result of a centralized computation should be computed. From this perspective, our problem is similar to finding an optimal-routing distributed strategy for the traditional data aggregation. In this case, a shortest path tree may be an option. However, to achieve our objective, only considering the problem from the perspective of theoretical computer science algorithms is not enough. The inherent additional characteristics of domain-specific algorithms, which can impose new constraints on the routing algorithm design, also need to be taken into consideration.

In a distributed algorithm for solving a least squares system, there are two features we need to consider. *First*, no matter what kind of data the sensors received, after being processed by local computation, the data size of the output is always a constant, which is determined by the number of parameters of interest, and at most equals to the size of one unit of raw measurement. From this perspective, the fewer sensors transmitting raw measurements the better. *Second*, the accuracy of the results from a centralized computation can only be guaranteed when the fisher information of each partial result obtained through local estimation is large enough. This condition is satisfied only when each sub-least squares system constructed by partial sensor measurements is well-conditioned, such that the precision of the final result can be guaranteed [Abe90]. That implies that a certain subset of sensors can sense and transmit raw measurements only and do nothing for the entire estimation when the condition number of the collected raw measurements is too large. *Finally*, the computation functions that each sensor executed is determined by the data types being received from its neighbors.

By considering these specific attributes of the least squares system, our objective is to design an optimal routing strategy in which the number of sensors transmitting raw measurements only is determined by the fisher information of the raw measurements, and adhere to the principle that fewer sensor nodes are better, as long as a given delay constraint is satisfied. Our problem can be formally formulated as follows:

Given:

- Wireless sensor network denoted as an undirected graph $G = (V, E)$, where V denotes the set of N sensor nodes and E denotes the set of edges representing the communication links between pairs of sensors.

- Each node k has access to t_{rows} scalar measurements d_k and a regression matrix $u_k \in \mathbb{R}^{(t_{\text{rows}} \times M)}$, where $k \in \{1, \ldots, N\}$.

- After local processing, the output data size is a constant which is equal to the number of parameters M.

Objective:

- To construct an optimal routing strategy for performing least squares estimation, such that the total amount of data transmitted in the entire network is minimized.

Subject to:

- The maximum number of hops from each nodes to the sink node is minimized, which is application-specific.

4.3 E^3: Distributed Least Squares Estimation

In this section, we will introduce the design of our new distributed algorithm to perform least squares estimation, referred to as E^3, in a progressive fashion. The goal is to achieve the performance of the centralized counterpart, but incurring a much lower transmission cost with a hop-count constraint.

4.3.1 Incremental recursive least squares estimation

Since our algorithm also adopts partial recursive processing, before introducing our new design, we first present a primer on how recursive strategy of distributed least squares works [SL06]. With incremental least squares estimation, the proposed algorithm requires the construction of a so-termed Hamiltonian path across sensors, and each node is allowed to communicate with its immediate neighbor in order to update the estimation result to a much better one. Each node only receives one existing estimate from its previous node along the path, and transmits the new estimates after local processing. The term "incremental" in [SL06] implies that, the least squares estimation of a "smallsized" regression matrix H is calculated first, which only involves the data from a few sensor nodes. Then the data from the remaining ones are incorporated incrementally into H, and each time measurements from a new sensor are added, the least squares estimation of the updated H is obtained by using only the previous estimation result and the newly added measurements. However, a Hamiltonian path across sensors for implementing incremental least squares is required, which has been proved to be a NP-hard problem. The number of hops in such a Hamiltonian path is determined by the size of the network, which is not practical in delay-sensitive applications.

Using the example illustrated in Sec. 3.2, if the Hamiltonian path exists, we can construct a path that visits each node exactly once as Fig. 4.1 shows. The estimation result using

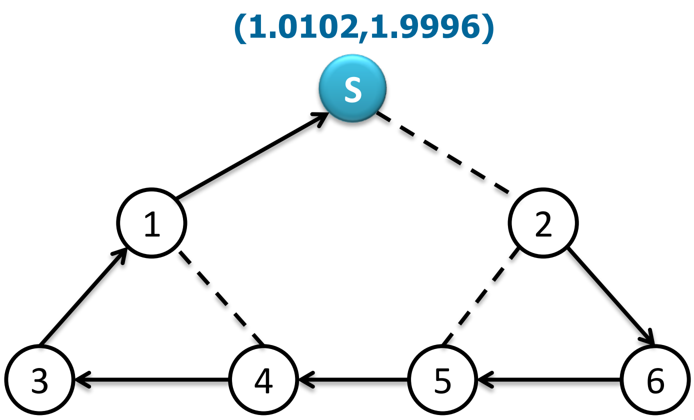


Fig. 4.1: Incremental recursive least squares estimation along a Hamiltonian path [SL06].

such a Hamiltonian path is as accurate as centralized computation, but the number of hops is equal to 6. When it comes to the worst case as [LCST12] illustrated, if the Hamiltonian path does not exist, the hop-count can double.

4.3.2 Distributed least squares estimation: a simple strategy as the s-tarting point

Inspired by the incremental recursive least squares algorithm, we first propose a simple strategy to perform distributed least squares estimation, as a starting point for designing E^3, which can satisfy the delay constraint of specific applications. In our simple strategy, the unknown parameters are updated along a *shortest path tree*. Along each path to the sink node, a partial incremental procedure which can update the previous results with the new measurements are implemented. From this perspective, our simple strategy can be regarded as a parallel version of the incremental recursive least squares algorithm along multiple paths. As a consequence, our simple strategy leads to a significant delay reduction. However, our simple strategy is not comparable to the incremental recursive least squares algorithm in terms of accuracy.

When it comes to updating estimation results using multiple previous estimates and raw measurements, recursive updating as shown in Eq. (4.2), which has been derived in [SL06], is not enough. Here μ_k denotes the local step size in each node [LS07]. The aggregation for multiple partial estimates is needed as shown in Eq. (4.3). For Eq. (4.3), [Abe90] clearly shows that the unknown parameters estimated are unbiased only when one of the following two conditions is satisfied: (1) each intermediate result has sufficient fisher information; or (2) the rows of initial values u must be no less than the number of parameters of interest. To a certain extent, these two unbiased conditions are equivalent. Thus, utilizing Eq. (4.2) and Eq. (4.3) to perform least squares estimation in a distributed manner can guarantee that the final results are able to converge to the global results once these conditions are satisfied.

$$\omega_k^0 = \omega_{k-1}^0 + \mu_k \times u_k(d_k - u_k \times \omega_{k-1}^0) \tag{4.2}$$

$$\omega_k^0 = \Sigma w \times \omega^0 \tag{4.3}$$

The estimation procedure is as follows: each node stores a pair of initial values, i.e, d_k and regression matrix u_k, $k \in \{1, \ldots, N\}$ which has been illustrated in Fig. 3.1. Unlike incremental recursive least squares, each node can receive initial measurement values or partial estimations from its neighboring nodes. Based on the data received, each sensor uses two different computation functions to compute a new estimate: Eq. (4.2) and Eq. (4.3). In our running example, the result of such computation is shown in Fig. 4.2.

However, the transmission cost is still not optimum because of the unbiased condition. This is especially the case when our simple strategy is used in some of domain-specific applications, such as damage detection in structural health monitoring. For practical use, we will need to design a better strategy for performing least squares estimation, taking into account specific requirements based on a specific global least squares estimation algorithm

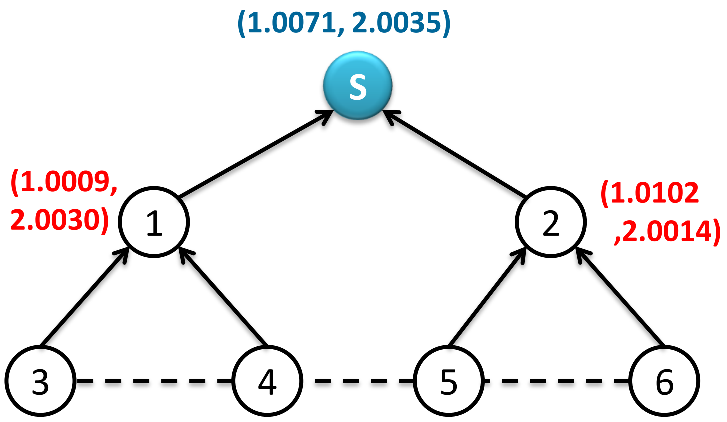


Fig. 4.2: Distributed least squares estimation: a simple strategy.

widely used in civil engineering [Avi01].

According to [Avi01], there is one fundamental requirement imposed by applications in the civil engineering domain that the rows of initial values included in each nodes are always smaller than the number of parameters needed to be estimated. However, according to [Avi01], the parameters of the civil structure need to be estimated $M = 5$, so here we can set the rows of initial values $t_{\text{rows}} = 4$ for simplicity. To ensure the convergence of our algorithm based on the unbiased estimation conditions as discussed above, the number of nodes has certain requirements that need to be included in the initial values of the regression matrix u_k, before the updating process can be started. In the context of civil engineering, we require that the number of sensor nodes included in the initial u should be no less than 2. This makes the size of the initial rows of u be no less than the number of parameters, leaving enough margin to ensure the fisher information of the initial estimation meets the unbiased requirement.

4.3.3 E^3: Our distributed algorithm

Before we introduce E^3, we first summarize the two existing strategies of distributed least squares estimation. The incremental recursive least squares estimation algorithm updates estimates along a Hamiltonian path, and each node computes its estimates using Eq. (4.2). Even though incremental least squares estimation can achieve the best possible accuracy and transmission cost, the delay is determined by the network size, rendering it less desirable in practical use with large sensor networks. In contrast, as a starting point, our simple strategy is to update the estimates along the shortest path tree, and the final result converges to the result of centralized computation. Even though the delay is minimized with the use of the shortest path tree, the transmission cost is not the optimum in the context of domain-specific applications in civil engineering.

Using our running example, as shown in Fig. 4.2, the computation of least squares estimation is updated along the shortest path tree. The total energy consumed is therefore equal to $3 \times 4 + 2 \times 2 = 16$, which has a 46% cost saving as compared to the centralized approach, even though the delay is still unchanged. We can see that our existing strategies are two extreme solutions to our problem. The first strategy based on the Hamiltonian path incurs the minimum transmission cost with the largest delay, while our simple strategy can achieve the minimum delay, though the transmission cost is not the optimum.

Can we achieve a balanced tradeoff between these two approaches so that the transmission cost is minimized with a delay constraint? The answer is affirmative. As discussed above, when performing partial estimation, the number of sensor nodes included in the initial u should be no less than 2. That means, when performing distributed least squares estimation in the shortest path tree, the leaf nodes only transmit the initial measurement values without any computation, while non-leaf nodes implement the two computation functions and transmit local estimates. The data size of the initial measurement values in each node is $t_{\text{rows}} \times (M+1)$, and the size of local estimates is M. If we assume the number of leaves in the shortest path tree is NumLeaf, the total transmission cost in the entire network is, therefore, shown in Eq. (4.4):

$$E = \text{NumLeaf} \times t_{\text{rows}} \times (M+1) + (N - \text{NumLeaf}) \times M \tag{4.4}$$

$$\min E = \text{NumLeaf} \times t_{\text{rows}} \times (M+1) + (N - \text{NumLeaf}) \times M \tag{4.5}$$

We wish to minimize such a total transmission cost, represented by Eq. (4.5). We can clearly see that if we can minimize the NumLeaf of the estimation tree, then we can find the optimum solution to our problem. Thus, our problem can be transformed to finding a spanning tree with a minimum number of leaf nodes, while the maximum hop count is no less than the constraint h. Our new distributed algorithm, E^3, is proposed to solve this problem by modifying the tree topology from the initial shortest path tree. The key idea is that, for each leaf node, only when it has sibling nodes and leaf neighbor nodes, its parent can be turned into its neighbor leaf node. Thus, the number of leave nodes can be reduced further. As such, E^3 utilizes a minimum-leaves spanning tree with a hop count constraint, which is shown in Algorithm 1.

Once the minimum leaves spanning tree is constructed, each leaf node only needs to transmit its initial values to its parent while non-leaf nodes implement Eq. (4.2) and Eq. (4.3) to update the intermediate results. The estimations are updated along the minimum leaves spanning tree. Using our algorithm to perform the estimation in a distributed manner in our running example, the routing strategy is shown in Fig. 4.3, with a total transmission cost of $3 \times 3 + 3 \times 2 = 15$, which leads to a 50% cost reduction in comparison to the

Algorithm 1 E^3: minimum leaves spanning tree with a hop count constraint.

Input: $G(V, E)$, the number of nodes N, the root node R, maximum hop count h
Output: Minimum leaves spanning tree with a hop count constraint

Construct a shortest path tree rooted in R
for each leaf node with a sibling k **do**
    for k's neighbor nodes m **do**
        if m is also a leaf node **then**
Save m's ID and its hop count to the sink node
        end if
    end for
    if k has a neighbor which is also leaf node i **then**
        if hop $< h$ after connecting k to i **then**
update k's parent to i and then number of leaves is reduced
        end if
    end if
end for

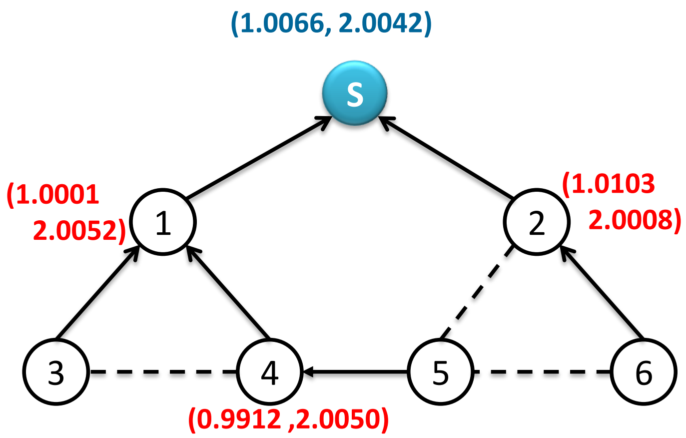


Fig. 4.3: Distributed least squares estimation using E^3.

centralized computation, and a 6% reduction from our simple strategy. The maximum hop count is 3 in our running example. In terms of accuracy, our algorithm has an estimation error of only 0.78%.

4.4 Performance Evaluation

4.4.1 Simulation

First, we use simulation to demonstrate the advantage of distributed least squares estimation in terms of transmission cost and estimation delay in sensor networks. We consider three different strategies of performing least squares estimation in a distributed manner.

1. **Strategy** 1 **:** all the measurements are transmitted to the sink node through the shortest path tree (i.e. centralized least squares estimation).

2. **Strategy** 2 **:** the incremental recursive least squares estimation along a Hamiltonian path [SL06].

3. **Strategy** 3 **:** our simple strategy described in section. 4.3.2, in which the unknown parameters are updated along the shortest path tree.

4. **Strategy** 4 **:** E^3, where the unknown parameters are updated along the path in the spanning tree with a minimum number of leaf nodes.

As discussed in Sec. 4.3, strategy 2 is required to find a Hamiltonian path to update the estimates. However, the Hamiltonian path may not exist. The best case of using strategy 2 is the case when the Hamiltonian path exists, then the total consumed energy can be calculated as $(N-1) \times M$ and its hop-count equals to $(N-1)$, both of which may become exceedingly high as the number of sensor nodes increases. However, if it does not exist, the total transmission cost can reach up to $2 \times (N-1) \times M$. Since the efficiency of strategy 2 is quite obvious, we do not consider it in our simulation.

In our simulation, two scenarios are created to evaluate their performance in different network densities and network sizes. Assume that N sensor nodes with the same communication range R are randomly deployed within a square area. We first fix the number of nodes $N = 100$, but gradually decrease the network density by decreasing R from 70 to 35. The area in this scenario is fixed to be 100×100. The maximum hop-count is fixed to be 25, which is the delay constraint. Then we maintain the network density at a certain level but gradually increase the network size by increasing N from 100 to 500. Note that to maintain the network density, when increasing N, the size of the deployment area also needs to be increased. In both scenarios, a total number of 100 simulations is performed and the average value of each strategy is calculated.

The results of the first scenario are illustrated in Fig. 4.4. First, it is easy to see that, the amount of data transmitted for the centralized least squares estimation is increasing with the decrease of R, when the number of nodes is fixed. Because much more relaying, for forwarding the raw measurements to the sink node, is required with the decrease of the communication range (note that this figure shows on the reverse x-axis). Different from strategy 1, Fig. 4.4 also shows that our approaches in strategy 3 and strategy 4 lead to a significant reduction in terms of communication cost with the decrease of R. There is no need to relay raw measurements in our approaches. Based on the discussions in Sec. 4.3, the leaf nodes only transmit raw measurements, while non-leaf nodes transmit partially estimated parameters whose data size is much smaller than that of raw measurements. E^3 is such a strategy where the unknown parameters are updated along the path in the minimum-leaves spanning tree with a hop-count constraint. Thus, it can save a significant amount of transmission cost by comparing with the shortest path tree approach (Strategy 3). Moreover, we can see from Fig. 4.4 that the more sparse the network, the more significant the advantage of E^3 in wireless communication.

Likewise, Fig. 4.5 compares the data to be transmitted in different strategies under various network sizes. It can be seen that the amount of data to be transmitted within strategy 3 and strategy 4 is much smaller than the centralized one (Strategy 1), and the

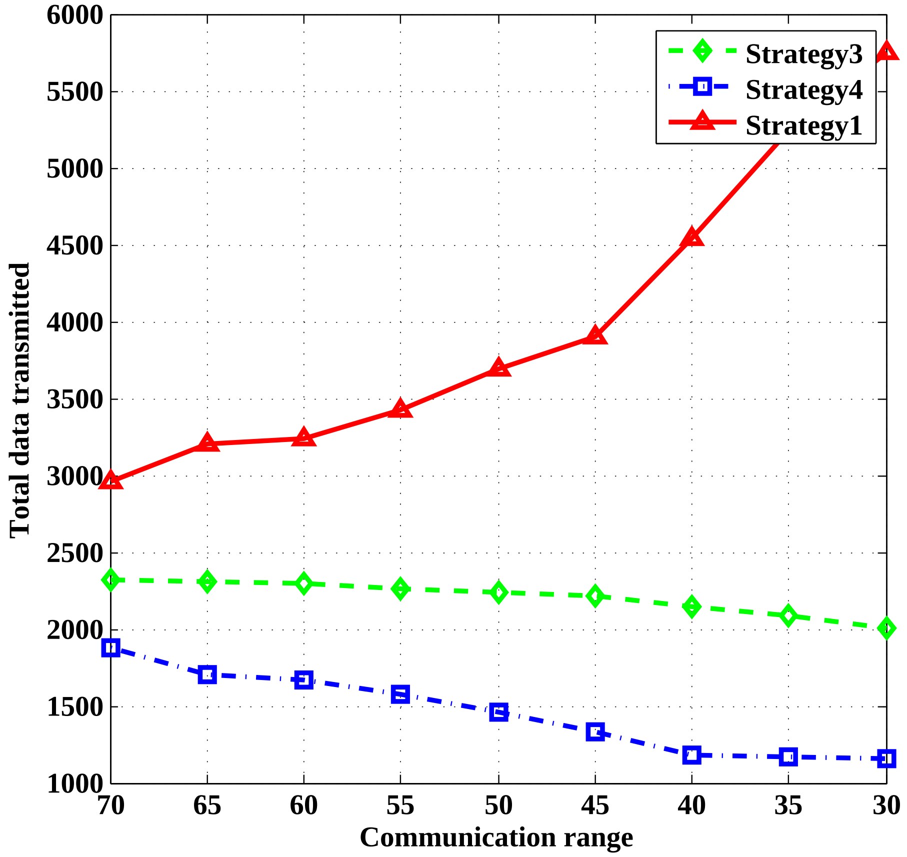


Fig. 4.4: The transmission cost with various network densities.

more sensor nodes in the network, this advantage becomes more remarkable. Due to the huge amount of data collection, strategy 1 is hard to put into practical use in some delay-sensitive applications in the context of wireless sensor networks. An important property that can be seen from Fig. 4.5 is that the advantage of our proposed method becomes more significant in a sparse network with a large number of sensor nodes, by comparison with the shortest path tree approach. This property is very favorable for structural health monitoring, since this matches the real condition when wireless sensor nodes are deployed to monitor the condition of large civil infrastructures.

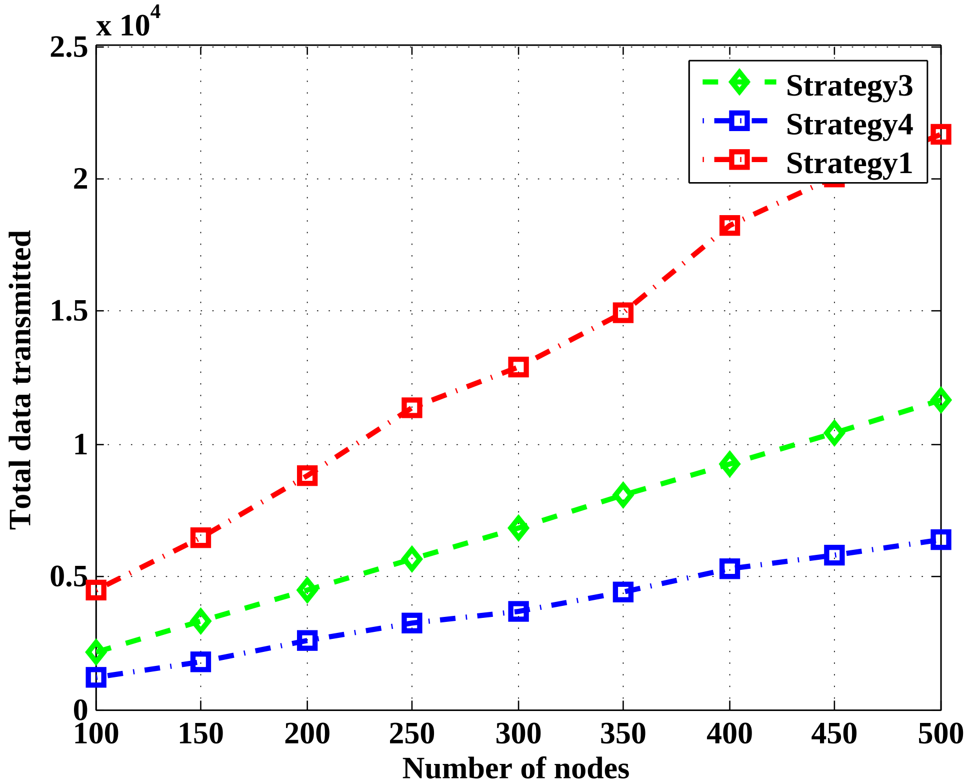


Fig. 4.5: The transmission cost with various network sizes.

4.4.2 Experiments in a building

To realistically evaluate the performance of our algorithms, we conduct structural damage detection experiments in a real environment to test the three strategies described in Sec. 4.4.1 (i.e. Strategy 1, Strategy 3, and Strategy 4). We deployed numbers of Structural Health Monitoring (SHM) motes in a building to measure the vibration data (see Fig. 4.6(a)). Then, we applied the least squares estimation method to estimate the natural frequencies, which can be used to analyze the health condition of the structure. The motes are specially designed by ourselves for the damage detection. The structural health monitoring motes, which contain a fairly powerful floating point digital signal processing processor

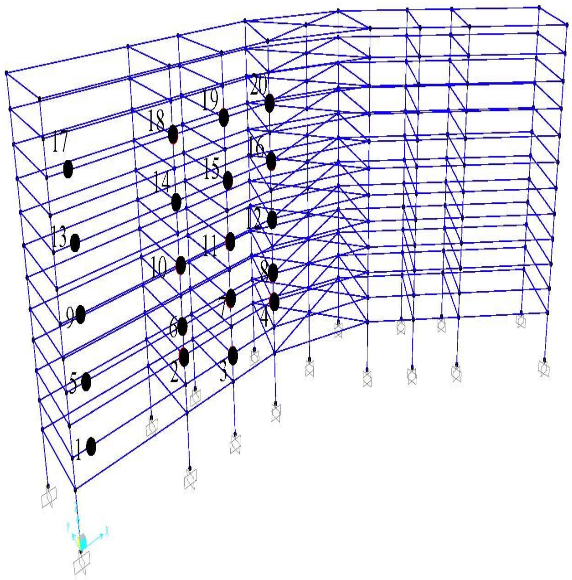
TMS320F2812 running at 150MHz, have the capability to implement sophisticated SHM computations, and are much faster than most off-the-shelf wireless sensor nodes.

The placement locations of the SHM motes and their numbering are shown in Fig. 4.6(b). The experiment system which is set up on No. 17 is shown in Fig. 4.7a. As illustrated in Fig. 4.7a, we use three highly sensitive external nodes KD1300 at each measurement location, to measure and record the vibration data of the building in three directions. The vibration signal recorded at KD1300 will be amplified, and then fed into a corresponding SHM mote where it is stored as 8-byte single-precision floating point format. For simplicity, we call the SHM motes connecting to the corresponding external nodes KD1300 the sampling motes.

In our experiment, the sampling motes at different locations do not communicate with each other directly, because they are not able to communicate directly when located in different rooms. To solve this problem, we use a particular SHM mote near the window of each location to act as a data collector as shown in the top figure of Fig. 4.7b. Once the sampling motes sample the vibrational data, the data will be transmitted to this collector first. In our context, we regarded the 20 collectors as independent wireless sensor nodes in the sensor network. With the objective of minimizing the communication cost in the entire network, we assume that the collectors have the local vibrational data already and only consider the wireless communication among collectors when we calculate the transmission cost.

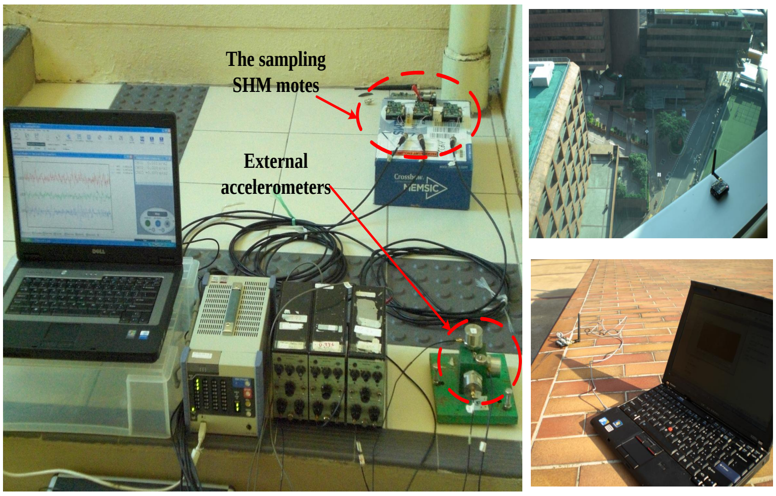
To test our strategies, we find out the sensor network topology of the 20 collector nodes based on the deployment using the collecting tree protocol (CTP) [GFJ$^+$09]. Then the topology information is transmitted to a gateway node which is connected to a laptop as shown in the bottom figure of Fig. 4.7b. The network topology constructed by the 20 collector nodes is illustrated in Fig. 4.8. By using the network topology, the laptop

(a)



(b)

Fig. 4.6: The building and measurement locations (a) The building (b) 20 measurement locations.



(a) (b)

Fig. 4.7: Experiment setup (a) Wired and wireless systems deployed at location 17 in the building (b) Top: a SHM mote deployed near the window as the collector node; Bottom: a gateway mote connected to a laptop computer.

constructs the shortest path tree rooted in No. 20 (for Strategy 1 and Strategy 3), or the spanning tree with minimum number of leaves also rooted in node No. 20 (for Strategy 4). For each node in the sensor network, its parent and children are then determined. And the relationship among collector nodes is then broadcast to the nodes by the laptop.

Then by utilizing the modified flooding time synchronization protocol (FTSP) [MKSL04], all of the sampling motes are synchronized. At a certain global time, all of the sampling motes start to sense with sampling rate of 1024Hz. The sampling procedure lasts for 50 seconds.

As discussed above, both Strategy 1 and Strategy 3 adopt the shortest path tree rooted at collector node 20 and, thus, they share the same routing strategy as illustrated in Fig. 4.8b. The difference between these two strategies is that in the former, the raw measurements are transmitted and collected along the tree to node 20, while in the latter the estimated parameters of least squares are updated along the shortest path tree. The routing strategy used by strategy 4 is shown in Fig. 4.8c. Along this routing, the parameters of interest are updated. In this figure, the estimation results will finally reach node 20, where the natural frequencies are calculated.

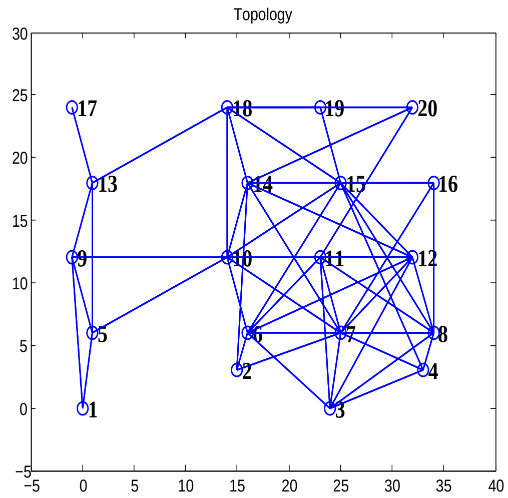
The theoretical amount of data transmitted in these three strategies are shown in Table 4.1. Updating the estimated parameters along the shortest path tree (Strategy 3) and along minimum-leaves tree (Strategy 4) are significantly better than delivering the measurements directly to the sink node (Strategy 1). Even though strategy 2 can lead to a minimum transmission cost, the Hamiltonian path does not always exist as discussed and it costs the maximum hop-count to reach the global results, which is 19 in the best case.

Strategies #	1	3	4
Theoretical data amount	40960	12323	7228
Maximum hop count	4	4	5

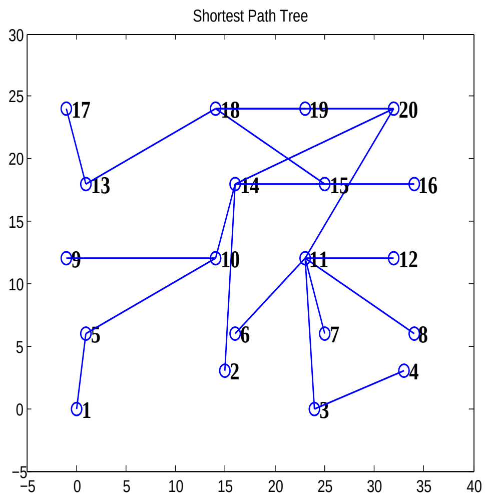
Table 4.1: Theoretical data amount and the actual maximum hop count in different strategies.

To satisfy the hop-count constraint, we also compare the maximum hop-count it takes with different strategies. With strategy 1, the hop-count it takes to finish the transmission of data is 4. While the number of hops it takes for updating the parameters in the shortest path tree (Strategy 3) and the minimum leaves tree (Strategy 4) are 4 and 5, respectively. With strategy 2, the hop-count it takes to update the unknown parameters in the Hamiltonian path is 19, and it is the longest one as compared to the other strategies.

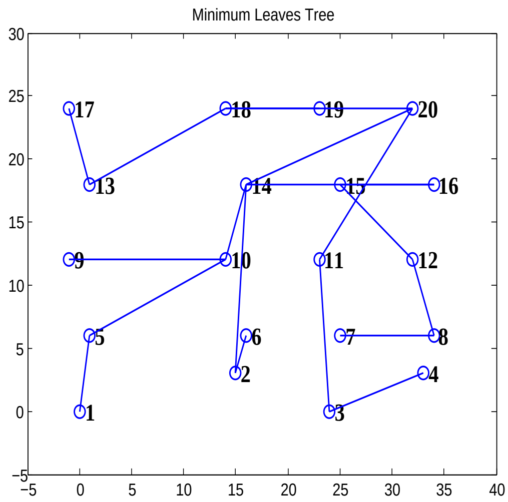
What's more, we illustrate that strategy 4 is able to achieve extremely accurate natural frequencies as was in the traditional centralized method. Natural frequencies calculated by using E^3 (Strategy 4) are compared with those calculated by using the traditional centralized method on the laptop computer connected with the gateway. The comparison results are shown in Table 4.2. The error in the frequency is calculated as the difference between the estimates on the SHM Mote and the PC. As shown in Table 4.2, the natural frequencies estimated in the SHM Mote and those on the PC are almost identical.



(a)



(b)



(c)

Fig. 4.8: (a) The network topology of collector nodes (b) The shortest path tree (with root 20) and (c) The minimum leaves tree (with root 20).

Mode #	1	2	3	4	5
The Centralized LSE	0.753	0.954	1.325	6.777	15.88
The Distributed LSE: E^3	0.752	0.961	1.333	6.805	15.79
Relative Error(%)	-0.133	0.734	0.604	0.413	-0.567

Table 4.2: Comparing accuracy of our approach to the centralized LSE. It can be seen that the identified natural frequencies of our approach have less than 1% identification error.

4.5 Summary

In this chapter, we study the problem of distributed least squares estimation in wireless sensor networks and propose a distributed approach, E^3, from a simple estimation strategy. By comparing with a simple strategy that updates the estimates along the shortest path tree, our approach is able to achieve the approximation quality of the centralized approach, yet using much lower wireless transmission costs and can satisfy the delay constraint. Corresponding to our proposed design framework, our strategy combines the advantages of T-AEc-U and T-ADe-U strategies. With simulation and a real experiment at test bed, the efficiency of E^3 in terms of transmission cost and delay has been demonstrated.

Chapter 5

Conclusions and Suggestions for Future Research

In this chapter, we conclude this thesis in Section 5.1 and outline some possible future works in Section 5.2.

5.1 Conclusions

Recent years, domain-specific applications, such as structural health monitoring, have been one of the main drivers that motivates the real-world deployment of wireless sensor networks. Due to their data-intensive and computation intensive nature, several unique challenging issues arise. It is typical for these applications to make heavy uses of least squares estimation as a foundation for their algorithms, which is a standard approach to compute the approximate solution of sets of equations in which there are more equations than unknowns. Due to the very limited amount of energy and computation power available on the sensors, it is imperative to design new algorithms to perform least squares estimation in a distributed fashion.

In this research, We mainly focus on algorithms design to solve least squares systems in wireless sensor networks. The distributed version should be able to achieve the same accuracy of the centralized counterpart under resource constraints (e.g. bandwidth, energy, computing capability, memory, etc).

Firstly, we proposed a 3 dimensional framework for general distributed least squares estimation algorithm design. Our framework can provide a basis for designing, analyzing and evaluating strategies to address parameters estimation issues using least squares estimation algorithms in wireless sensor networks. In the 3D framework, we proposed three design aspects(aggregation structure, performance metric and communication pattern) of designing distributed least squares estimation, and then we study the existing works from the design aspects and then discuss advantages and disadvantages correspondingly.

Secondly, we proposed E^3, a tree-based distributed algorithm for least squares estimation, which can reach the objective of minimizing the energy consumption incurred during communication, while observing constraints on application-specific communication delays. And then we conduct simulations and real experiments for structural damage detection in a real environment to test the performance of our algorithms. Compared to previous works, we show that E^3 maintains the same level of estimation precision while incurring much lower energy costs.

At last, we address that our 3D framework can facilitate the design, classification and evaluation of the current distributed least squares estimation strategies in sensor networks by utilizing our tree-based strategy.

5.2 Suggestions for Future Research

We close this thesis by providing some suggestions for future research. Specifically, we mainly focus on least squares estimation in this thesis. However, least square estimation is only one kind of computing methodology. We believe that the following aspects which consider a general distributed computing architecture are worth further investigations.

5.2.1 Distributed Computing Architecture

Given a sophisticated signal processing algorithm $T(D)$ which applies computation T on data D consisting data from sensors across the whole network, to design the distributed version of $T(D)$ must utilize the computation capability of each sensor. In addition, we wish to carry out the computation on the route when each node sends its data to the sink. This indicates that each sensor should carry out a sub-computation task $T_i(D_i)$ using its own or received data and transmit the result to its parent. When the data flow reaches the sink, the obtained result is the same with that of $T(D)$ but with much less cost compared to the existing 'data collection then centralized computation'. To achieve this goal, we first investigate some general computation architectures. Then given a certain $T(D)$, we try to find some types of key computation functions by which$T(D)$ can be decomposed and choose the appropriate computation architecture accordingly. At last, based on the computation architecture and the graph of the deployed sensors, the optimal routing is then obtained.

In particular, for any data sets $\{D_i\} \subseteq D$ and S_i obtained by $T(\{D_i\}) = S_i$, $B1$ and $B2$ should satisfy that:

$$B1(S_i, \{D_j\}) = T(\{D_i\} \bigcup \{D_j\}) \tag{5.1}$$

$$B2(S_i, S_j, \ldots) = T(\{D_i\} \bigcup \{D_j\} \bigcup \ldots) \tag{5.2}$$

Task 1.1: Firstly, for a given centralized signal processing algorithm $T(D)$ which applies computation T on data D, we will find out some types of building-block sub-tasks by which $T(D)$ can be constructed and from which, choose the best computation architecture shown in Fig. 3.3.A sensor node can receive two types of data: raw data and intermediate results. For $T(D)$ to be decomposed within a network, we must construct one or two types of building-block sub-tasks: (1)$B1(S_i, \{D_j\}) = S_k$, and (2)$B2(S_i, S_j \ldots) = S_p$. Where $\{D_i\}$ is raw data set from a set of sensors,$S_i, S_j, S_k \ldots$ are the intermediate results which can be obtained either by applying T on data from part of the sensors ($i.e.T(\{D_i\}) = S_i$), or from the output of and $B2(S_i, S_j \ldots)$. It can be seen that $B1$ is designed to update an intermediate result by incorporating raw data, and $B2$ is used to fuse a set of intermediate results.

Task 1.2: Secondly, given $T(D)$ and the building-block sub-tasks $B1$, $B2$, we will find out extra conditions (if there is any) that the results of the distributed version using $B1$ and $B2$ is the same with the centralized counterpart. It should be noted that, it is quite possible that $B1/B2$ can only satisfy Eq.5.1/ Eq.5.2 at certain conditions. For example, to use the DAC method to combine two intermediate results x_i and x_j respectively from $A_i * x_i = b_i$ and$A_j * x_j = b_j$, both x_i and x_j must have large enough Fisher information (i.e. $B2(S_i, S_j) = T(\{D_i\} \bigcup \{D_j\}$ only when both S_i and S_j have enough Fisher information). These requirements must be considered when we are using a computation architecture shown in Fig. 3.3.

Task 1.3: Thirdly, we should design the optimal routing according to the objective functions and constraints. Given a computation task $T(D)$, the chosen computation architecture, and the graph of deployed sensors $G = (V, E)$, we need to find out the optimal routing for the sensor node in the network. The routing is optimal mainly in the sense that the energy consumption and delay should be minimized.

References

[Abe90] Jonathan S Abel. A divide and conquer approach to least-squares estimation. *Aerospace and Electronic Systems, IEEE Transactions on*, 26(2):423–427, 1990.

[Avi01] Peter Avitabile. Experimental modal analysis. *Sound and vibration*, 35(1):20–31, 2001.

[BA] James Brownjohn and Emin Aktan. Improving resilience of infrastructure: the case of bridges.

[BMS11] Alexander Bertrand, Marc Moonen, and Ali H Sayed. Diffusion bias-compensated rls estimation over adaptive networks. *Signal Processing, IEEE Transactions on*, 59(11):5212–5224, 2011.

[CLS08] Federico S Cattivelli, Cassio G Lopes, and Ali H Sayed. Diffusion recursive least-squares for distributed estimation over adaptive networks. *Signal Processing, IEEE Transactions on*, 56(5):1865–1877, 2008.

[CLW12] Kang-Ming Chang, Shin-Hong Liu, and Xuan-Han Wu. A wireless semg recording system and its application to muscle fatigue detection. *Sensors*, 12(1):489–499, 2012.

[CS11] Federico S Cattivelli and Ali H Sayed. Analysis of spatial and incremental lms processing for distributed estimation. *Signal Processing, IEEE Transactions on*, 59(4):1465–1480, 2011.

[DFPS96] Scott W Doebling, Charles R Farrar, Michael B Prime, and Daniel W Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. Technical report, Los Alamos National Lab., NM (United States), 1996.

[GEAdlVJGQ11] Antonio Gomez-Exposito, Ali Abur, Antonio de la Villa Jaen, and Catalina Gomez-Quiles. A multilevel state estimation paradigm for smart grids. *Proceedings of the IEEE*, 99(6):952–976, 2011.

[GFJ+09] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection Tree Protocol. In *Embedded Networked Sensor Systems, ACM International Conference on* . ACM, November 2009.

[Gol] Gaetan Kerschen-Jean-Claude Golinval. Experimental modal analysis.

[KPKK07] Jong-Won Kwon, Yong-Man Park, Sang-Jun Koo, and Hiesik Kim. Design of air pollution monitoring system using zigbee networks for ubiquitous-city. In *Convergence Information Technology, 2007. International Conference on*, pages 1024–1031. IEEE, 2007.

[KSS13] Goutham Kamath, Lei Shi, and Wen-Zhan Song. Component-average based distributed seismic tomography in sensor networks. In *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pages 88–95. IEEE, 2013.

[KZG92] Moshe Kam, QIANG Zhu, and WS Gray. Optimal data fusion of correlated local decisions in multiple sensor detection systems. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(3):916–920, 1992.

[LCST12] Xuefeng Liu, Jiannong Cao, Wen-Zhan Song, and Shaojie Tang. Distributed sensing for high quality structural health monitoring using wireless sensor networks. In *Real-Time Systems Symposium, IEEE International Conference on*, pages 75–84. IEEE, 2012.

[LL06] Jerome P Lynch and Kenneth J Loh. A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*, 38(2):91–130, 2006.

[LS06] Cassio G Lopes and Ali H Sayed. Distributed adaptive incremental strategies: formulation and performance analysis. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, volume 3, pages III–III. IEEE, 2006.

[LS07] Cassio G Lopes and Ali H Sayed. Diffusion least-mean squares over adaptive networks. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, volume 3, pages III–917. IEEE, 2007.

[LS08] Cassio G Lopes and Ali H Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *Signal Processing, IEEE Transactions on*, 56(7):3122–3136, 2008.

[MCP+02] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In

Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pages 88–97. ACM, 2002.

[MKSL04] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *Embedded Networked Sensor Systems, ACM International Conference on*, pages 39–49. ACM, 2004.

[MPK08] Henry Medeiros, Johnny Park, and Avi Kak. Distributed object tracking using a cluster-based kalman filter in wireless camera networks. *Selected Topics in Signal Processing, IEEE Journal of*, 2(4):448–463, 2008.

[Ren98] RA Renaut. A parallel multisplitting solution of the least squares problem. *Numerical linear algebra with applications*, 5(1):11–31, 1998.

[SL06] Ali H Sayed and Cassio G Lopes. Distributed recursive least-squares strategies over adaptive networks. In *Signals, Systems and Computers, Fortieth Asilomar Conference on*, pages 233–237. IEEE, 2006.

[SL07] Ali H Sayed and Cassio G Lopes. Distributed processing over adaptive networks. In *Signal Processing and Its Applications, IEEE International Symposium on*, pages 1–3. IEEE, 2007.

[SSK$^+$] Lei Shi, Wen-Zhan Song, Goutham Kamath, Guoliang Xing, and Xuefeng Liu. Distributed least-squares iterative methods in networks: A survey.

[SSX$^+$] Lei Shi, Wen-Zhan Song, Mingsen Xu, Qingjun Xiao, Jonathan M Lees, and Guoliang Xing. Imaging seismic tomography in sensor network.

[T$^+$93] John N Tsitsiklis et al. Decentralized detection. *Advances in Statistical Signal Processing*, 2:297–344, 1993.

[XV96] Julie C Xia and Amitabh Varshney. Dynamic view-dependent simplification for polygonal models. In *Visualization'96. Proceedings.*, pages 327–334. IEEE, 1996.

[YWM05] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1214–1217. IEEE, 2005.

[Zom96] Albert YH Zomaya. *Parallel and distributed computing handbook.* McGraw-Hill, Inc., 1996.