WIRELESS SENSOR NETWORK LIFETIME ANALYSIS AND ENERGY EFFICIENT TECHNIQUES

by

HAINING SHU

A Ph.D Dissertation Proposal

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2007

To my parents

For their endless love.

ACKNOWLEDGEMENTS

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my advisor, Dr. Qilian Liang. Throughout my doctoral work Dr. Liang gave me valuable research guidance and encouraged me to develop in independent thinking and research skills. I had no experience of doing graduate level research and knew nothing about wireless sensor networks when I first joined the group. Dr. Liang instructed me from the begining very patiently and exhibited gradually to me the beauty of research work. The mentorship and support of Dr. Liang is of most importance in completion of this dissertation. In the past four years, I learned from Dr. Liang not only the knowledge and research skills, but also the spirit of perseverance and devotion toward research, teaching and life. This admirable spirit of Dr. Liang has inspired me throughout my doctoral work and will continue to inspire me in the rest of my life.

I am also very grateful for having an exceptional doctoral committee and wish to thank Dr. W. Alan Davis, Dr. Jean Gao, Dr. Wei-Jen Lee and Dr. Kamisetty R. Rao for their continual support and encouragement.

Many friends have helped me stay sane through these difficult years. I greatly value their friendship and I deeply appreciate their belief in me. I would like to acknowledge the members of Wireless Communications and Networking Research Group at UTA and my old friends in UTD.

Last but most importantly, I would like to express my heart-felt gratitude to my parents and my older sister, Cuifeng, Huaquan and Jianjuan. None of this would have been possible without the love and patience of my families.

July 16 , 2007

ABSTRACT

WIRELESS SENSOR NETWORK LIFETIME ANALYSIS
AND ENERGY EFFICIENT TECHNIQUES

Publication No. _____

Haining Shu, Ph.D.

The University of Texas at Arlington, 2007

Supervising Professor: Qilian Liang

Wireless microsensor networks is one of the most important technologies for the 21st century. In distributed sensor networks energy-aware techniques are used to reduce energy consumption. Various sensor network applications have taken energy efficiency into consideration.

This thesis report focuses on a new approach based on fuzzy logic systems to analyze the lifetime of a wireless sensor network. It demonstrates that a type-2 fuzzy membership function(MF), i.e., a Gaussian MF with uncertain standard deviation (std) is most appropriate to model a single node lifetime in wireless sensor networks. This research studies two basic sensor placement schemes: square-grid and hex-grid. Two fuzzy logic systems(FLSs): a singleton type-1 FLS and an interval type-2 FLS are designed to perform lifetime estimation of the sensor network. Simulation results show that the FLS offers a feasible method to analyze and estimate the sensor network lifetime and the interval type-2 FLS in which the antecedent membership functions are modeled as Gaussian with uncertain std outperforms the singleton type-1 FLS.

In the later chapters, two energy efficient techniques in wireless sensor networks are presented: fuzzy optimization for distributed sensor deployment and spectrum efficient coding scheme for correlated non-binary sources in wirless sensor networks.

For the sensor deployment topic, it is shown that given a finite number of sensors, optimizing the sensor deployment will provide sufficient sensor coverage and ameliorate the quality of communications. We apply fuzzy logic systems to optimize the sensor placement after an initial random deployment. We use the outage probability due to co-channel interference to evaluate the communication quality. Fenton-Wilkinson method is applied to approximate the sum of log-normal random variables. Our algorithm is compared against the existing distributed self-spreading algorithm. Simulation results show that our approach achieves faster and stabler deployment and maximizes the sensor coverage with minimum energy consumption. Outage probability, as a measure of communication quality gets effectively decreased in our algorithm but it was not taken into consideration in the distributed self-spreading algorithm.

In the case of correlated binary sources, distributed source coding has been literally studied in information theory. However, data sources from real sensor networks are normally non-binary. We proposed a spectrum efficient coding scheme for correlated non-binary sources in sensor networks. Our approach constructs the codeword cosets for the interested source, taking advantage of statistical characters of the distinct observations from sensor nodes. The coset leaders are then transmitted via the channel and decoding is performed with the available side information. Simulations are carried out over independent and identically distributed (i.i.d) Gaussian sources and data collected from Xbow wireless sensor network test bed. Simulation results show that the proposed scheme performs at 0.5 - 1.5 dB from the Wyner-Ziv distortion bound.

The wireless sensor technology can be applied to many real world applications. In this dissertation report, we present two applications when the sensor technology is used in the multi-target data fusion and underwater target positioning.

For the multi-target data fusion, we consider the decision fusion of Rayleigh fluctuating targets in multi-radar sensor networks. Decision fusion and data fusion in Wireless Sensor Networks (WSNs) has been widely studied in order to save energy. Radar system as a special sensor network, when implemented for battlefield surveillance, faces bandwidth constraint in real-time applications instead of energy restriction. A reliable detection of multiple targets in clutter is perhaps the most important objective in such an echo-location system. In this work, we study the decision fusion rules of multiple fluctuating targets in multi-radar (MT-MR) sensor networks. The MT-MR decision fusion problem is modeled as a multi-input multi-output (MIMO) system. We assume that each radar makes binary decision for each target from the observation, i.e. if the target is present or not. We derive our *MIMO fusion* rules based on the target fluctuation model and compare against the optimal likelihood ratio method (LR), maximum ratio combiner (MRC) and equal gain combiner (EGC). Simulation results show that the *MIMO fusion* rules approach the optimal-LR and outperforms MRC and EGC at high signal to clutter ratio (SCR).

In Chapter 6, we present a silent positioning scheme termed as UPS for underwater acoustic sensor networks. UPS relies on the time-difference of arrivals measured locally at a sensor to detect range differences from the sensor to four anchor nodes. These range differences are averaged over multiple beacon intervals before they are combined to estimate the 3D sensor location through trilateration. UPS requires no time-synchronization and provides location privacy at underwater vehicles/sensors whose locations need to be determined. To study the performance of UPS, we model the underwater acoustic channel as a modified Ultra Wide Band (UWB) S-V model: the arrival of each path cluster

and paths within each cluster follow double Poisson distributions, and the multipath channel gain follows a Rician distribution. Based on this channel model, we perform both theoretical analysis and simulation study on the position error of UPS under acoustic fading channels. The obtained results indicate that UPS is an effective scheme for underwater vehicle/sensor self-positioning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Research on sensor networks was originally motivated by military applications. Starting around 1980, networked microsensors technology has been widely used in military applications. One example of such applications is the Cooperative Engagement Capability (CEC) developed by the U.S.Navy. This network-centric warfare consists of multiple radars collecting data on air targets [1]. Other military sensor networks include acoustic sensor arrays for antisubmarine warfare such as the Fixed Distributed System (FDS) and the Advanced Deployable System (ADS), and unattended ground sensors (UGS) such as the Remote Battlefield Sensor System (REMBASS) and the Tactical Remote Sensor System (TRSS).

Although nowadays the majority of sensors are still wired, wireless sensors provide significant advantages over wired sensors. Two main problems within wired sensor networks - cost and delays in deployment are tackled when low-cost sensors and communication networks become available.

Wireless sensor networks have recently come into research notability because they developed many other non-military applications, from environment and habitat monitoring, to industrial process control, to infrastructure security [2] and automation in transportation. One networked sensing experiment on Great Duck Island [3] provides a small lens into an expansive future of such applications. The experiment was conducted by a team of computer engineers from the University of California, Berkeley. To date, 190 wireless sensors have been deployed on a small island 10 miles off the coast of Maine

to study the nesting behaviors of petrels. Biologists are now monitoring the petrels on the island from their offices, browsing data from sensors linked by satellite.

A wireless sensor network consists of a certain amount of small and energy constrained nodes. Basic components of a sensor node include a single or multiple sensor modules, a wireless transmitter-receiver module, a computational module and a power supply module. Such networks are normally deployed for data collection where human intervention after deployment, to recharge or replace node batteries may not be feasible, resulting in limited network lifetime. Most applications have pre-specified lifetime requirements. For instance the petrels monitoring application in [3] has a lifetime requirement of at least 9 months. Thus estimation of lifetime of such networks prior to deployment becomes a necessity. Prior work on evaluating lifetime have considered networks where sensor nodes are randomly deployed. [4] and [5] gives the upper bound on lifetime that any network with the specified number of randomly deployed nodes, source behavior and energy can reach while [6] discusses the upper bounds on lifetime of networks with cooperative cell based strategies. Network lifetime of fixed deployment schemes are recently studied in [7]. Jain and Liang observed that in wireless sensor networks, a single node lifetime behaves like a normal Gaussian distribution which brings the first light of exploring the network lifetime behavior given the knowledge of nodes lifetimes [7].

Research has been done to alleviate the energy consumption and extend network lifetime in wireless sensor networks, from hardware design of individual sensor to routing and topology construction of the whole network. Among which, one distinct technology for energy-efficient wireless sensor networks is distributed source coding (DSC) [8,9]. DSC was proposed to encode the correlated sensor readings separately, i.e. sensors encoding the readings do not communicate with each other. After the distributed encoding, the compressed data is sent to a central hub node for joint decoding. Further research on

this topic demonstrated that convolutional codes [10], Turbo and LDPC codes [11, 12] performed well in distributed compression for sensor networks. All these approaches are based on binary distributed sources with refined correlation to each other. However, in a practical sensor network or even in a lab test bed of wireless sensor network, the distributed deployed sensors have very rough readings which can hardly be fitted into the above binary compressing schemes.

Another challenging topic in energy-efficient wireless sensor networks is sensor deployment. The deployment of sensors varies with different applications. A number of applications require the placement of sensors at desired locations like data collection [13] and infrastructure security [14], where critical area, buildings and facilities are monitored by a network of sensors placed adequately. For such placement-friendly applications, sufficient knowledge of the environment is assumed to be available before deployment is carried out.

In other applications where prior knowledge of the environment can not be obtained, sensors may have to be randomly air-dropped and human intervention after deployment to recharge or replace node batteries may not be feasible. Mobile sensors are practically desirable in this situation because they have the capability to move around and re-adjust their positions for high quality communication and better coverage and surveillance [15]. However mobile sensor deployment itself is an energy consuming process because of the motion and communication between sensors. An efficient sensor re-deployment scheme is a necessity to save energy resources and improve the quality of communications.

Some prior research proposed sensor deployment strategies based on virtual forces for target localization [16], [17], [18]. One example of the virtual force concept was presented in [18]. The pair-wise interaction between sensor nodes is governed by two kinds of virtual forces - one causes the nodes to repel each other to improve their coverage and

the other is an attractive force that prevents the nodes from losing connectivity. Later Cheng *et al.* [19] formulated a constrained multivariable nonlinear programming problem to determine both the locations of the sensor nodes and data transmission pattern. In [20] and [21], Heo and Varshney developed a distributed self-spreading algorithm (DSSA) and an intelligent deployment and clustering algorithm (IDCA) for sensor deployment. Recently, a voronoi diagram (VD)-based deployment algorithm was included in [22]. All the above algorithms have made lots of efforts to formulate the virtual forces, however none of which can handle well the uncertainties with the random move and unpredictable oscillation in sensor deployment. For the purpose of stability and convergence, various parameters or constraints such as oscillation limit, stable status [20–22], and number of neighbors [18] have to be imposed to avoid excessive sensor oscillation.

The concept of wireless sensor networking can be applied to the traditional radar system. Radar as a powerful sensor system, has been employed for the detection and location of reflecting objects such as aircraft, ships, vehicles, people and the natural environment. By radiating energy into space and detecting the echo signal reflected from an object or target, the radar system can determine the presence of a target. Furthermore, by comparing the received echo signal with the transmitted signal, the location of a target can be determined along with other target related information [23].

Conventional radar system operates as a pure independent entity. While in a resource-constrained WSN, such detached operation may lead to deteriorated performance and waste of limited resources. Collaborative signal and information processing over the network is a very promising area of research and is related to distributed information fusion [24]. Important technical issues include the degree of information sharing between sensors and how sensors fuse the information from other sensors. Processing data from more sensors generally results in better performance but also requires more communication resources. Similarly, less information is lost when communicating infor-

mation at a low level (e.g., raw data), but requires more bandwidth. Therefore, it is a tradeoff between system performance and resource utilization in collaborative information processing and data fusion.

A lot of prior research in data fusion are based on the assumption of lossless communication, i.e., the information sent from local sensors is perfectly recovered at the fusion center. For example, in [25] and [26], Vashney *et. al* investigated the optimum fusion rules under the conditional independence assumption. Other papers [27, 28] addressed the problem of distributed detection with constrained system resources, most of which provided the solutions to optimize sensor selection. However, this lossless communication assumption is not practical for many WSNs where the transmitted data suffers from channel fading and multi-user interference. In another hand, decision fusion with non-ideal communication channels are studied at both fusion center level [29, 30] and at the sensor level [31, 32]. In [30], Thomopoulos and Zhang derived the optimal thresholds by assuming a simple binary symmetric channel between sensors and the fusion center. Their method is quite simple but requires global knowledge of the entire system. In [29], channel-aware decision fusion rules have been developed using a canonical distributed detection system where binary decisions from multiple parallel sensors are transmitted through fading channels to a fusion center. Later, Lin *et. al* [33] have extended the channel aware decision fusion rules to more realistic WSN models that involve multi-hop transmissions. The above results, however, are mostly obtained based on one target or one event detection which is not applicable to multi-target situations. Furthermore, in a radar sensor system, when clutter, the unwanted echoes from the natural environment is much larger than the receiver noise, detection can be quite difficult from that when the noise is dominant.

Recently, underwater acoustic sensor networks have attracted lots of attention. UnderWater Acoustic Sensor Networks (UWA-SNs) consists of a variable number of sensors

and vehicles to perform collaborative monitoring tasks over a given area. The main motivation for UWA-SNs is their relative ease of deployment since they eliminate the need for cables and they do not interfere with shipping activities. UWA-SNs are envisioned to enable applications for environmental monitoring of physical and chemical/biological indicators, tactical surveillance, disaster prevention, undersea exploration, assisted navigation, etc.

Location discovery for underwater vehicles/sensors is nontrivial in the oceanic medium. Propagation delays, motion-induced Doppler shift, phase and amplitude fluctuations, multipath interference, etc., are all significant factors in location measurement. The well-known Global Positioning System (GPS) receivers, which may be used in terrestrial systems to accurately estimate the geographical locations of sensor nodes, do not work properly in underwater [34]. Some localization schemes based on received signal strength (RSS), time of arrival (ToA), or angle of arrival (AoA), could be used. Nevertheless, the bandwidth constraint, sensor mobility, and unpredicted variation in channel behavior make many of these approaches inaccurate or unapplicable [35].

This thesis is organized as follows. Chap 2 addresses the issue of lifetime analysis and estimation for wireless sensor networks in which the sensor nodes are deployed at desired locations. Our approach is entirely different from all prior research. In stead of trying out various probability basis, we propose to apply an interval type-2 fuzzy logic system (FLS) for lifetime analysis and estimation in a wireless sensor network. We demonstrate that a type-2 fuzzy membership function(MF), i.e., a Gaussian MF with uncertain variance is most appropriate to model a single node lifetime in wireless sensor networks. Two FLSs: a singleton type-1 FLS and an interval type-2 FLS are constructed for lifetime analysis and estimation. Simulations are implemented on two basic placement schemes: square-grid and hex-grid. We believe that these two schemes can serve as basis

for evaluating more complex schemes for their lifetime performance prior to deployment and help justify the deployment costs.

Chap 3 applies type-1 fuzzy logic systems (FLSs) to handle these uncertainties in distributed sensor deployment. Instead of attempting to formulate the virtual forces, FLSs are proposed to re-deploy the sensors. Each individual mobile sensor uses a FLS to self-adjust its location. For a single sensor node, neighboring nodes' location is the only information needed to make the movement decision. Therefore the deployment scheme based on FLSs is a fully-distributed approach. After applying FLSs, exhaustive move and unpredictable oscillation is efficiently avoided and fast deployment is achieved. As a result, the entire sensor network survives for longer lifetime and the quality of communication in terms of outage probability is

Chap 4 investigates the spectrum efficient coding scheme for correlated non-binary sources in wireless sensor networks. Our approach attempts to provide a solution to Chief Executive Officer (CEO) problem. The goal of the CEO problem is to recover as much information as possible about the actual event from the noisy observations, while minimizing the total information rate. We propose to exploit the statistical characters of real sensor readings before constructing codeword cosets. From the approximate Gaussian readings, Lloyd-Max quantization is applied to minimize the mean square distortion. To save communication spectrum, a coset encoder is designed to reduce the transmitted bits based on the probability distribution of quantized values. We show that source encoding can be completed in a fully distributed way. Each sensor encodes its own readings without knowing what the other sensors have measured. Our work differs from previous ones not only in the non-binary sources but in proposing a practical coset encoding scheme for real senor readings.

Chap 5 considers the decision fusion of Rayleigh fluctuating targets in multi-radar sensor networks. Decision fusion and data fusion in Wireless Sensor Networks (WSNs) has

been widely studied in order to save energy. Radar system as a special sensor network, when implemented for battlefield surveillance, faces bandwidth constraint in real-time applications instead of energy restriction. A reliable detection of multiple targets in clutter is perhaps the most important objective in such an echo-location system. In this work, we study the decision fusion rules of multiple fluctuating targets in multi-radar (MT-MR) sensor networks. The MT-MR decision fusion problem is modeled as a multi-input multi-output (MIMO) system. We assume that each radar makes binary decision for each target from the observation, i.e. if the target is present or not. We derive our *MIMO fusion* rules based on the target fluctuation model and compare against the optimal likelihood ratio method (LR), maximum ratio combiner (MRC) and equal gain combiner (EGC). Simulation results show that the *MIMO fusion* rules approach the optimal-LR and outperforms MRC and EGC at high signal to clutter ratio (SCR).

Chap 6 introduces an application of underwater acoustic sensors. A silent positioning scheme termed as UPS is presented for underwater acoustic sensor networks. UPS relies on the time-difference of arrivals measured locally at a sensor to detect range differences from the sensor to four anchor nodes. These range differences are averaged over multiple beacon intervals before they are combined to estimate the 3D sensor location through trilateration. UPS requires no time-synchronization and provides location privacy at underwater vehicles/sensors whose locations need to be determined. To study the performance of UPS, we model the underwater acoustic channel as a modified Ultra Wide Band (UWB) S-V model: the arrival of each path cluster and paths within each cluster follow double Poisson distributions, and the multipath channel gain follows a Rician distribution. Based on this channel model, both theoretical analysis are performed and a simulation study on the position error of UPS under acoustic fading channels. The obtained results indicate that UPS is an effective scheme for underwater vehicle/sensor self-positioning.

Simulation results are presented in each chapter respectively. Chap 7 summarizes the main results of the research work and outlines the future research directions.

CHAPTER 2

WIRELESS SENSOR NETWORK LIFETIME ANALYSIS

2.1 Preliminaries

A sensor network is designed to perform high-level information processing tasks such as target detection and tracking. Applications of sensor networks are wide ranging and can vary significantly in modes of deployment, application requirements, sensing models and quality of services (QoS). In this section, the key assumptions related to our approach are discussed, then formally characterize the basic modes of sensor placement in this study followed by the concept of coverage, connectivity and network lifetime.

2.1.1 Assumptions and Notations

The approach of network lifetime analysis developed in this paper is based on the following assumptions about the wireless sensor networks:

- Sensor nodes are placed in a two-dimensional field. Nodes operate with very limited energy resources which shape the aspects of the node performance. For instance the node's processing ability, sensing and communication range is limited. It is assumed all sensor nodes originally deployed are identical in power configuration and functionality.
- Sensing and communication between sensor nodes is done through radio links. An assumption is made that the radio range for a node is a disk of radius r around the node where r is taken as the same for all nodes. A sensor node can detect or sense any target or event within its *sensing range*, denoted by R_s. A *communication range* R_c is also defined which is the distance beyond which the transmitted signal

is received with signal to noise ratio (SNR) below the acceptable threshold level. In this paper, the communication range R_c is assumed to be equal to the sensing range R_s.

- To be more energy efficient, direct communication between any pair of sensor nodes is allowed only if the Euclidean distance r between them satisfies $r \leq R_c$. Such a pair of nodes are called neighboring nodes. Since communication within communication range R_c is by broadcast, all immediate neighbors hear what a node transmits. Communication between non-neighboring nodes is achieved via peer-to-peer communication and may consist of several short hops to relay the transmitted message.
- The maximum allowable distance between two sensor nodes who can communicate directly, R_{max}, is equal to the communication range R_c. In study it follows that $R_{max} = R_c = R_s$. A sensor network is said to be deployed with *minimum density* when the Euclidean distance between all neighboring nodes is $r = R_{max}$.

2.1.2 Basic Modes of Sensor Placement

Sensor placement directly affects power resource management and background data processing which is carried out with various sensed data in distributed sensor networks. An intelligent sensor placement enlarges the field coverage, facilitates the operation of sensors (detecting, communication, data aggregation etc.) and reduces the excessive communications in fulfilling a task.

The simplest placement modes involve uniform or regular placement of sensor nodes such that each sensor node in the network has the same number of neighbors. A sensor placement mode that regulates two neighbors per sensor node has been described in [36]. According to [7], the *Square-grid* and *Hex-grid* placement modes are illustrated in Fig. 2.1(a) and (b) respectively. Square-grid mode in Fig. 2.1(a) shows that each

sensor node in the network has four neighbors and in the Hex-grid mode in Fig. 2.1(b), the number of neighbors for every sensor node is three. It is believed that these three elementary placement modes [7, 36] can serve as a basis for other placement schemes, because a placement scheme of any complexity can be decomposed into two-neighbor, three-neighbor and four-neighbor groups.

The distance r in Fig. 2.1(a) and (b) denotes the communication range. Sensor nodes in both Square-grid and Hex-grid are equidistant from their respective neighbors. According to the assumption, both grids are deployed with minimum density[1].

2.1.3 Coverage and Connectivity

Coverage and connectivity are two important performance metrics of networks and hence a discussion on them becomes imperative before the lifetime of the network can be defined.

Coverage scales the adequacy with which the network covers the sensor field. A sensor with sensing range R_s is said to cover or sense a circular region of radius R_s around it. If every point in the sensor field is within distance R_s from at least one sensor node, then the network is said to provide complete or 100% coverage.

Various levels of coverage are acceptable depending on the application. In critical applications, complete coverage is required at all times. Any loss of coverage leads to a sensing gap in the field. Such gaps cause breach of security in case of surveillance applications. Also, in applications which require data with high precision a sensing gap leads to inaccuracies. For such networks any loss of coverage renders the network nonfunctional. While in some other applications a small loss of coverage may be acceptable.

[1]The *Hex-grid* is observed to have lower density than the *Square-grid*. Rough calculation shows that with 36 nodes deployed, the network with *Square-grid* covers an area of approximate $25r^2$, and the *Hex-grid* covers an area of approximate $48r^2$, almost double that of the *square grid*.

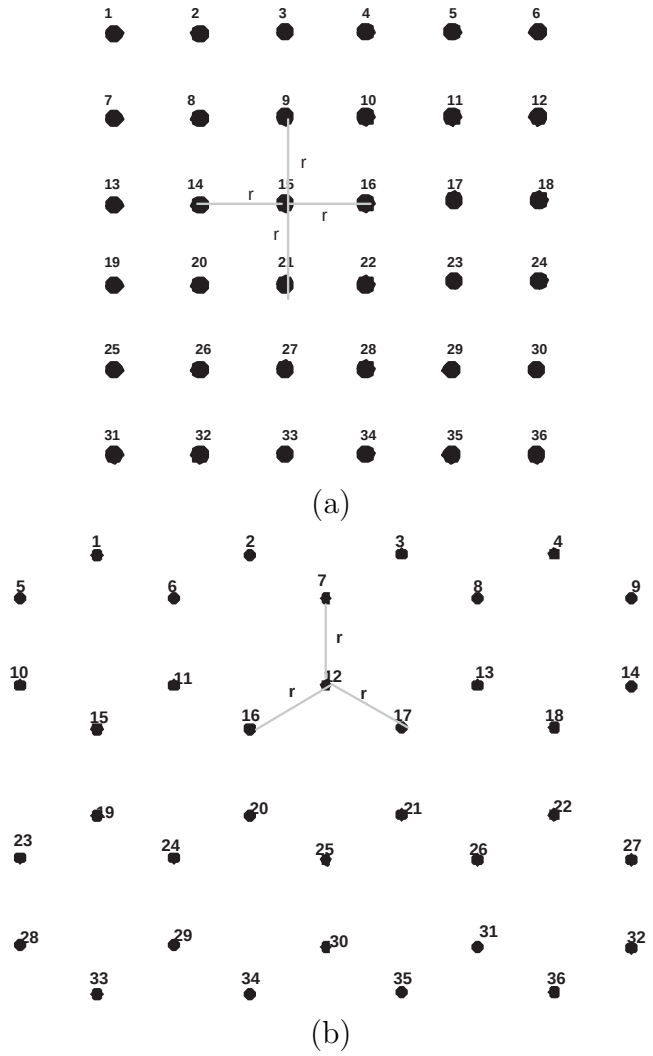


Figure 2.1. Two Placement Modes: (a) Square-Grid (b) Hex-Grid.

Connectivity scales the adequacy with which nodes are able to communicate with their peers. One of the strengths of sensor networks arises from their ability to aggregate data collected from different sensor nodes. This requires adequate communication between sensor nodes. Any node should be able to communicate with any other node for proper functioning of the network. If a large number of nodes fail due to lack of energy, a part of the network may get completely disconnected from the rest. If a large number of nodes fail due to lack of energy, a part of the network may get completely

disconnected from the rest. In the study only 100% connectivity is acceptable and the network fails with any loss of connectivity. An example of a sensor placement scheme that concentrates mainly on coverage as its parameter of interest can be found in [37], where a sensor placement algorithm for grid coverage has been proposed.

In the analysis it is required the network to provide complete coverage and connectivity. Equal importance is given to both parameters and declare the network nonfunctional if either of them falls below their desired levels.

2.1.4 Network Lifetime

The basic definition of lifetime, or more precisely the post-deployment active lifetime of a network is the cumulative active time measured from deployment until network failure. Based on the levels of coverage and connectivity required to deem a network functional, network failure can be interpreted in different ways. Since only complete coverage and connectivity are acceptable to us, network failure corresponds to the first loss of coverage or connectivity.

This paper concentrates on finding the minimum lifetime of a network, the worst case scenario. To be able to evaluate this minimum lifetime, we need to know the lifetime of a single sensor node, the minimum number of node failures that cause network failure, and the positional relationship [2] between the failed nodes.

Consider the Square-grid and the Hex-grid networks deployed with minimum density. Both networks survive the failure of a single node without loss of either connectivity or coverage implying that the minimum number of node failures that can lead to network failure is greater than one. Now looking into the Square-grid network in Fig. 2.2(a), apparently failure of nodes 20 and 21 causes loss of coverage. Similarly failure of nodes 20

[2]Positional relationship between two nodes can be that the two nodes are diagonal, adjacent or completely unrelated.

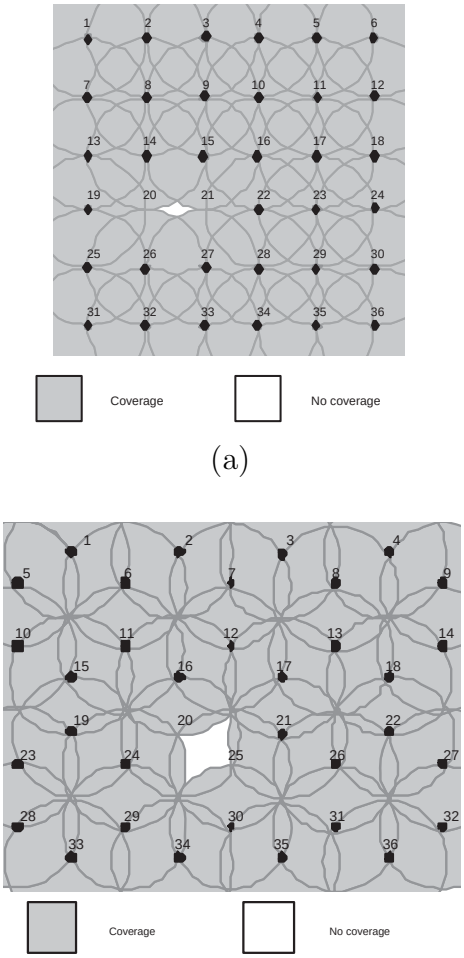


Figure 2.2. Loss of coverage examples: (a)Square-grid. (b)Hex-grid.

and 25 in the Hex-grid network brings the coverage down to less than 100%. It comes to a conclusion that the failure of any two neighboring nodes causes loss of coverage and hence network failure.

Thus the minimum number of node failures that cause network failure is two and these two nodes must be adjacent to each other (neighbors). A network may undergo multiple node failures and still be connected and covered if any of the failed nodes are not

neighbors. But the absolute minimum number of node failures that can cause network failure is two.

2.2 Introduction to Fuzzy Sets and Interval Type-2 Fuzzy Logic Systems

2.2.1 Introduction to Type-2 Fuzzy Sets

The concept of type-2 fuzzy sets was introduced by Zadeh [38] as an extension of the concept of an ordinary fuzzy set, i.e., a type-1 fuzzy set. Type-2 fuzzy sets have grades of membership that are themselves fuzzy [39]. A type-2 membership grade can be any subset in $[0, 1]$ – the *primary membership*; and, corresponding to each primary membership, there is a *secondary membership* (which can also be in $[0, 1]$) that defines the possibilities for the primary membership. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity. Type-2 fuzzy sets handle linguistic uncertainties, as typified by the adage "words can mean different things to different people." A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal, "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner [40]".

Figure 2.3 shows an example of a type-2 set. The domain of the membership grade corresponding to $x = 4$ is also shown. The membership grade for every point is a Gaussian type-1 set contained in $[0, 1]$, we call such a set a "Gaussian type-2 set". When the membership grade for every point is a crisp set, the domain of which is an interval contained in $[0, 1]$, such type-2 sets are called "interval type-2 sets" and their membership grades "interval type-1 sets" [41]. Interval type-2 sets are very useful when we have no other knowledge about secondary memberships.

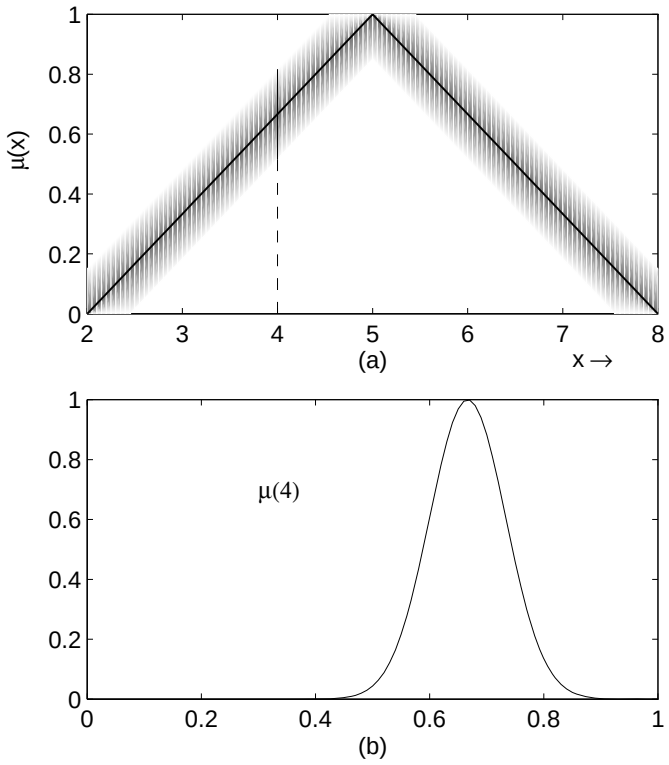


Figure 2.3. (a) Gaussian type-2 set. (b) The secondary memberships.

An interval type-2 fuzzy set can be represented by its *upper* and *lower* membership functions (MFs) [42]. An upper MF and a lower MF are two type-1 MFs which are bounds for the footprint of uncertainty (the union of all primary membership grades) of an interval type-2 MF. The upper MF is a subset which has the maximum membership grade of the footprint of uncertainty; and, the lower MF is a subset which has the minimum membership grade of the footprint of uncertainty.

An overbar (underbar) is used to denote the upper (lower) MF. For example, let $\tilde{\mathrm{F}}_k^l(x_k)$ denote the type-2 MF for the kth antecedent of the lth rule, then the upper and lower MFs of $\mu_{\tilde{\mathrm{F}}_k^l}(x_k)$ are $\overline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)$ and $\underline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)$, respectively, so that

$$\mu_{\tilde{\mathrm{F}}_k^l}(x_k) = \int_{q^l \in [\underline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k), \overline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)]} 1/q^l \tag{2.1}$$

where $\int$ denotes the union of individual points of each set in the continuum.

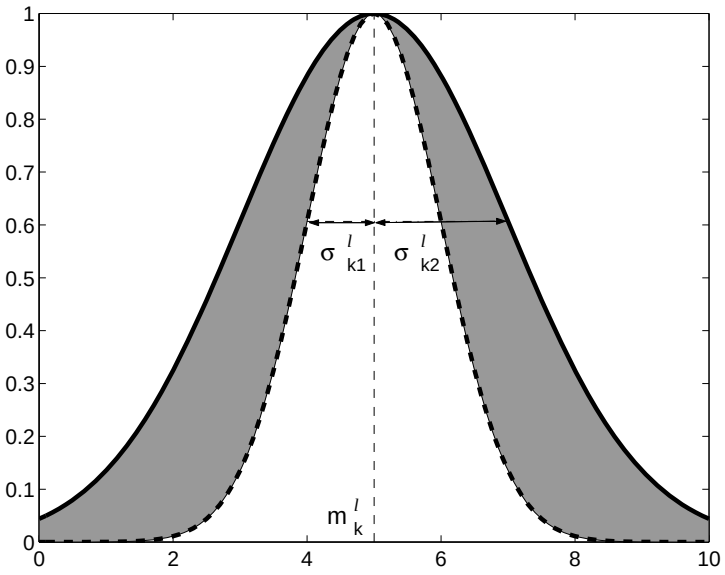


Figure 2.4. Type-2 Gaussian MF with uncertain standard deviation.

Example 1: Gaussian Primary MF with Uncertain Standard Deviation

Consider the case of a Gaussian primary MF having a fixed mean, m_k^l, and an uncertain standard deviation that takes on values in $[\sigma_{k1}^l, \sigma_{k2}^l]$, i.e.,

$$\mu_k^l(x_k) = \exp\left[-\frac{1}{2}(\frac{x_k - m_k^l}{\sigma_k^l})^2\right], \quad \sigma_k^l \in [\sigma_{k1}^l, \sigma_{k2}^l] \tag{2.2}$$

where: $k = 1, \ldots, p$; p is the number of antecedents; $l = 1, \ldots, M$; and, M is the number of rules. The upper MF, $\overline{\mu}_k^l(x_k)$, is (see Fig. 2.4)

$$\overline{\mu}_k^l(x_k) = \mathcal{N}(m_k^l, \sigma_{k2}^l; x_k), \tag{2.3}$$

and the lower MF, $\underline{\mu}_k^l(x_k)$, is (see Fig. 2.4)

$$\underline{\mu}_k^l(x_k) = \mathcal{N}(m_k^l, \sigma_{k1}^l; x_k) \tag{2.4}$$

□

This example illustrates how to define $\overline{\mu}$ and $\underline{\mu}$, so that it is clear how to define these membership functions for other situations (e.g., triangular, trapezoidal, bell MFs).

2.2.2 Introduction to Type-2 Fuzzy Logic Systems: An Overview

Figure 2.5 shows the structure of a type-2 FLS [41]. It is very similar to the structure of a type-1 FLS [43]. For a type-1 FLS, the *output processing* block only contains the defuzzifier. We assume that the reader is familiar with type-1 FLSs, so that here we focus only on the similarities and differences between the two FLSs.

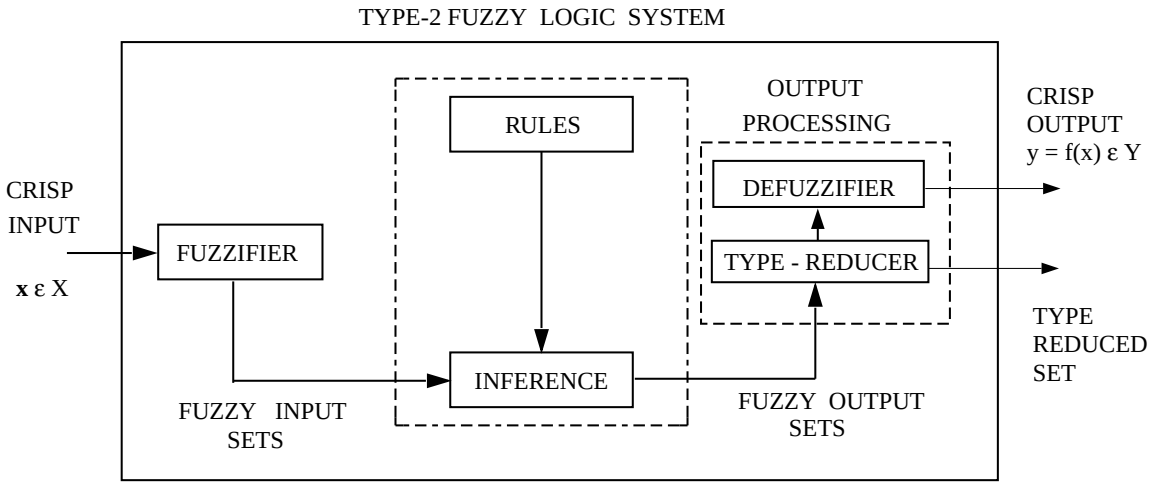


Figure 2.5. The structure of a type-2 FLS..

The fuzzifier maps the crisp input into a fuzzy set. This fuzzy set can, in general, be a type-2 set.

In the type-1 case, it generally has "IF-THEN" rules, where the lth rule has the form "R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l", where: x_is are inputs; F_i^ls are antecedent sets ($i = 1, \ldots, p$); y is the output; and G^ls are consequent sets. The distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important while forming rules; hence, the structure of the rules remains exactly the same in the type-2 case, the only difference being that now some or all of the sets involved are of type-2; so, the lth rule in a type-2 FLS has the form "R^l : IF x_1 is $\tilde{\mathrm{F}}_1^l$ and x_2 is $\tilde{\mathrm{F}}_2^l$ and $\cdots$ and x_p is $\tilde{\mathrm{F}}_p^l$, THEN y is $\tilde{\mathrm{G}}^l$".

In the type-2 case, the inference process is very similar to that in type-1. The inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. To do this, one needs to find unions and intersections of type-2 sets, as well as compositions of type-2 relations.

In a type-1 FLS, the defuzzifier produces a crisp output from the fuzzy set that is the output of the inference engine, i.e., a type-0 (crisp) output is obtained from a type-1 set. In the type-2 case, the output of the inference engine is a type-2 set; so, "extended versions" (using Zadeh's Extension Principle [38]) of type-1 defuzzification methods was developed in [41]. This extended defuzzification gives a type-1 fuzzy set. Since this operation takes us from the type-2 output sets of the FLS to a type-1 set, this operation was called "type-reduction" and the type-reduced set so obtained was called a "type-reduced set" [41]. To obtain a crisp output from a type-2 FLS, we can defuzzify the type-reduced set.

General type-2 FLSs are computationally intensive, because type-reduction is very intensive. Things simplify a lot when secondary membership functions (MFs) are interval sets (in this case, the secondary memberships are either 0 or 1). When the secondary MFs are interval sets, the type-2 FLSs were called "interval type-2 FLSs". In [42], Liang and Mendel proposed the theory and design of interval type-2 fuzzy logic systems (FLSs). They proposed an efficient and simplified method to compute the input and antecedent operations for interval type-2 FLSs, one that is based on a general inference formula for them. They introduced the concept of upper and lower membership functions (MFs) and illustrate their efficient inference method for the case of Gaussian primary MFs. They also proposed a method for designing an interval type-2 FLS in which they tuned its parameters.

In an interval type-2 FLS with *singleton fuzzification* and meet under minimum or product t-norm, the result of the input and antecedent operations, F^l, is an interval type-1 set, i.e., $F^l = [\underline{f}^l, \overline{f}^l]$, where $\underline{f}^l$ and $\overline{f}^l$ simplify to

$$\underline{f}^l = \underline{\mu}_{\tilde{\mathrm{F}}_1^l}(x_1) \star \ldots \star \underline{\mu}_{\tilde{\mathrm{F}}_p^l}(x_p) \tag{2.5}$$

and

$$\overline{f}^l = \overline{\mu}_{\tilde{\mathrm{F}}_1^l}(x_1) \star \ldots \star \overline{\mu}_{\tilde{\mathrm{F}}_p^l}(x_p) \tag{2.6}$$

where x_i $(i = 1, \ldots, p)$ denotes the location of the singleton.

In this paper, center-of-sets is used for type-reduction, which can be expressed as:

$$Y_{\cos}(Y^1, \cdots, Y^M, F^1, \cdots, F^M) = [y_l, y_r] = \int_{y^1} \cdots \int_{y^M} \int_{f^1} \cdots \int_{f^M} 1 \Big/ \frac{\sum_{i=1}^{M} f^i y^i}{\sum_{i=1}^{M} f^i} \tag{2.7}$$

where $Y_{\cos}$ is an interval set determined by two end points, y_l and y_r; $f^i \in F^i = [\underline{f}^i, \overline{f}^i]$; $y^i \in Y^i = [y_l^i, y_r^i]$, and Y^i is the centroid of the type-2 interval consequent set $\tilde{\mathrm{G}}^i$; and, $i = 1, \ldots, M$. Because $Y_{\cos}$ is an interval set, we defuzzify it using the average of y_l and y_r; hence, the defuzzified output of an interval type-2 FLS is

$$f(\mathbf{x}) = \frac{y_l + y_r}{2} \tag{2.8}$$

2.2.3 Applications of Interval Type-2 Fuzzy Logic Systems

Liang and Mendel have developed theory and design methods for the most useful kind of type-2 fuzzy logic system (FLSs), interval type-2 FLSs [42], and have applied them to a number of very important applications, such as

1. Fading channel equalization [44] and co-channel interference elimination [45]. The channel states in a fading channel or channel with co-channel interferences are uncertain, and they validated that an interval type-2 fuzzy set, Gaussian primary membership function with uncertain mean, can be used to represent such uncertainties.

2. Network video traffic modeling and classification [46]. MPEG variable bit rate (VBR) traffic are very bursty. They validated that the I, P, and B frame sizes are log-normal with fixed mean and uncertain variance, so an interval type-2 fuzzy sets can be used to model the bursty video traffic and an interval type-2 fuzzy logic system with such type-2 fuzzy set are demonstrated performing much better than a Bayesian classifier.
3. Connection admission control for ATM network [47]. Connection admission control is actually a decision making problem. Different factors such as incoming real-time video/audio packet sizes, non-real time packet sizes, the buffer sizes are uncertain. They applied an interval type-2 fuzzy logic to handle these uncertainties.

2.3 Modeling Node Lifetime with Gaussian Membership Functions

Though applications of sensor networks vary with task requirements, sensor nodes retain the basic elements within any sensor network. Behavior of individual sensor in a manner determines the network performance. As we have discussed in Section 2.1.3, failure of sensor nodes due to lack of energy may render the network functioning improperly or the whole network completely breaking down. Thus it becomes significantly important to characterize a single node behavior before going into the network layer. A distinct identity of sensor nodes and wireless sensor network is its constrained energy resources which is widely measured by the node and network lifetime. In this section, we study the lifetime manner for single sensor node and explore its probability characters from real network data.

Since wireless sensor nodes are severely energy constrained due to their compact form, hardware design and protocol approaches for different layers must take energy efficiency into account to increase the lifetime of sensor networks. However, a fundamental question - "what is the nature of sensor network lifetime?" has not been answered yet.

Because the lifetime of each individual node is not constant but random variables, it follows that the network lifetime is also a random variable. Recently, Jain and Liang [7] showed that in a wireless sensor network where the workloads are very well-balanced, a single node lifetime behaves the nature of normal Gaussian distribution. Their observation was also justified by the knowledge of probability and random process in [7]. Fig 2.6 illustrates the real node lifetime distribution in a hex-grid sensor network.

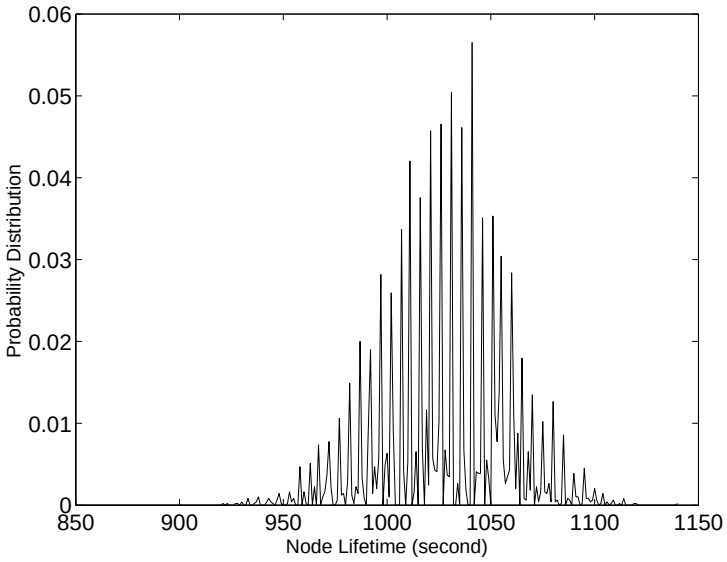


Figure 2.6. A single node lifetime distribution in a hex-grid sensor network.

In the study, the first interest is to set up a precise membership function (MF) for the single node lifetime. From the original data of single node lifetime shown in Table 2.1, we decomposed the whole data sets into seven segments, and computed the mean μ_i and standard deviation σ_i of node lifetime for each segment, $i = 1, 2, ...7$. The mean μ and standard deviation σ for the entire data set were also computed. We are also interested to know which value - mean μ_i or standard deviation σ_i varies more. We first normalized the mean μ_i and standard deviation σ_i of each segment using μ_i/μ and σ_i/σ. Then we

Table 2.1. Mean and STD values for seven segments and the entire node lifetime

Node Lifetime Data	Mean	Std
Segment 1	1027.4	30.182
Segment 2	1028.9	29.819
Segment 3	1026.3	30.798
Segment 4	1028.7	30.917
Segment 5	1028	29.944
Segment 6	1027	29.975
Segment 7	1027.9	30.306
Entire Data Set	1027.7	30.292
Normalized STD	0.00082783	0.013105

computed the standard deviation of their normalized values σ_m and σ_{std}. Results are presented in the end of Table 2.1.

From the last row of Table 2.1, we see that $\sigma_m << \sigma_{std}$ which means standard deviation σ_i varies much more than the mean value μ_i. Therefore we conclude that if the single node lifetime follows normal Gaussian distribution, it is most appropriate to be modeled as a Gaussian MF with uncertain standard deviation. This result justifies the use of the Gaussian MFs to model single node lifetime in Section 2.4.

2.4 Sensor Network Lifetime Analysis Using Interval Type-2 FLSs

In Section 2.3, we came to a conclusion that in wireless sensor networks, the single node lifetime can be well modeled by a type-2 Gaussian MF with uncertain variance. This result offers a truly original approach to probe into the trait of network lifetime. Referring to [4, 6, 7], prior research on sensor network lifetime are mostly heuristic or application and protocol driven. In this paper, the fuzzy logic systems are proposed to sensor network lifetime analysis, more specifically to evaluate or estimate network lifetime using interval type-2 fuzzy logic systems (FLSs). The network lifetime evaluation system is named as *Type-2 Fuzzy Logic Lifetime Evaluator(FLLE2).*

From the overview of type-2 fuzzy logic systems (FLSs) in Section 2.2.2, we know that in a type-2 FLS, when p inputs are applied to the system, the inference engine computes the output set corresponding to each rule. After the unique type-reduction operation, the defuzzifier computes a crisp output from these rule output sets. We next expatiate the input-output parameters, type-2 antecedent and consequent MFs and fuzzy rules precisely designed for evaluating sensor network lifetime.

2.4.1 Antecedent and Consequent Memebership Functions

Let N be the number of sensor nodes deployed in a sensor network. All sensor nodes are initially configured with the same battery level. After a period of time during which several tasks (detecting, target tracking and communication etc.) were carried out by different sensors, the network is left with sensor nodes at various battery levels. It is assumed the sensor network of interest at this point remains at complete coverage and connectivity and all sensor nodes are still alive. The problem of evaluating network lifetime is formulated like this:

Given a set of data representing the various battery levels of all N alive sensor nodes, denoted by $p(1)$,...,$p(N)$, estimate the cumulative alive time T of the network measured from the point of interest until network failure, where $p(i)$ is the current battery level of sensor node i.

The $N \times 1$ values of battery levels, $p(1)$,...,$p(N)$ are taken as input to the type-2 fuzzy logic evaluator (FLLE2) and the alive time T of the network measured from the point of interest until network failure is taken as the output of FLLE2.

One antecedent is depicted as "the remaining battery level of sensor node i" and the consequent as "the cumulative network alive time from the point of interest until network failure" The linguistic variables to represent the antecedent are divided into

three levels: *high*, *moderate* and *low* and the consequent is divided into five levels: *very high*, *high*, *moderate*, *low* and *very low*.

Antecedent and consequent membership functions are chosen based on the result in Section 2.3. A type-2 Gaussian MF with uncertain standard deviation is illustrated in Fig 2.4. In this study, all antecedents and consequents use the same type of MFs and the MF parameters are initialized consistently.

2.4.2 Rules Design

We apply *reliability theory* from control system to design the fuzzy rules. In this section, we first treat the basics of reliability theory then demonstrate how this knowledge is extracted for rules design with two simple examples.

2.4.2.1 Basics of Reliability Theory

For the sensor network lifetime issue studied in this paper, reliability theory provides a feasible method to design fuzzy rules. To understand this, The reliability block diagram (RBD) is introduced. RBD is a graphical representation of the components of the system, and provides a visual representation of the way components are reliability-wise connected. Thus the effect of the success or failure of a component on the system performance can be evaluated.

Consider a system with two components. If this system is such that a single component failure can render the system nonfunctional, then it is said that the components are reliability-wise connected in series. If the system fails only when both its components fail, then it is said that the components are reliability-wise connected in parallel. Note that the physical connection between the component may or may not be different from their reliability-wise connection. The RBD's for both cases are given in Fig. 2.7. Any

complex system can be realized in the form of a combination of blocks connected in series and parallel.

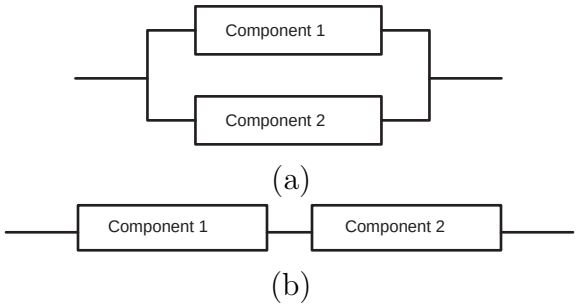


Figure 2.7. (a)Parallel RBD (b)Serial RBD.

2.4.2.2 Rules Design Using Reliability Theory

In the analysis, the wireless sensor network is the system under consideration and the sensor nodes are the components of the system. We detail below how to design rules with the knowledge of reliability theory referring to the two basic *RBDs* in Fig. 2.7. It is assumed that the workload among all components (sensor nodes) are very well-balanced.

Example 2: Set up Fuzzy Rules for the Parallel System in Fig. 2.7(a)

In the parallel system, crisp logic claims that the system (network) fails only when both components (sensor nodes) fail. In fuzzy logic systems, the rules can be set up as one example shown bellow:

IF the remaining battery level of component 1 (sensor node 1) is *high* and the remaining battery level of component 2 (sensor node 2) is *moderate*, THEN the lifetime of the system (network) is *very high*.

Example 3: Set up Fuzzy Rules for the Series system in Fig. 2.7(b)

In the series system, crisp logic claims that the system (network) fails when either component fails. In fuzzy logic systems, the rules can be set up as one example shown bellow:

IF the remaining battery level of component 1 (sensor node 1) is *low* and the remaining battery level of component 2 (sensor node 2) is *moderate*, THEN the lifetime of the system (network) is *low*.

Note that the parallel and series systems are the essential ways to model two sensor nodes. A wireless sensor network consisting of multiple sensor nodes can be hierarchically represented in the reliability block diagram.

2.4.3 Case Studies: Rules Design

2.4.3.1 Square-Grid Sensor Network

As defined in Section 2.1.4, the minimum network lifetime is the time to failure of any two neighboring nodes. We know that the failure of any single node does not cause network failure. The failure of any node coupled with the failure of any of its neighbors causes network failure. Using this definition the *RBD* is built for the square-grid as shown in Fig 2.8.

Fig 2.8 shows the *RBD* block for a single node in the network. A node can be modeled in two ways depending on its position in the sensor field. This distinction based on its position is made due to a simple observation that nodes at the right edge of the sensor field (region-2) do not have any right neighbor (node b) as opposed to nodes in region-1. Also, nodes at the bottom edge of the sensor field (region-2) do not have a bottom neighbor (node c) as opposed to the nodes in region-1. Note that as every node in a square-grid, node a has four neighbors, but its relationship with only two neighbors is modeled in its *RBD* block. This is because the relationship with the other two neighbors

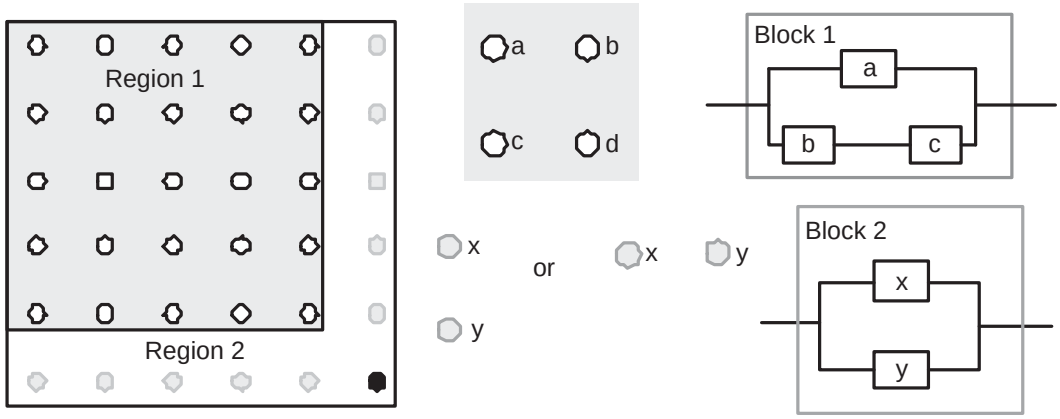


Figure 2.8. RBD of a single node in a square grid.

will be modeled when their *RBD* blocks are constructed. If this is not followed then the relationship between every node-neighbor pair will be modeled twice.

We abstract three antecedents from the *RBD* of block-1 in Fig 2.8.

- Antecedent 1 - The remaining battery level of node a.
- Antecedent 2 - The minimum remaining battery level of node b and c.
- Antecedent 3 - The remaining battery level of node d.

The consequent is depicted as "minimum network lifetime". The linguistic variables to represent the antecedent are divided into three levels and the consequent is divided into five levels as detailed in 2.4.1. Total $27 = 3^3$ rules (3 antecedents and each has 3 fuzzy sub-sets) are set up for square-grid block-1. Table 2.2 gives the complete 27 designed rules.

Two antecedents are chosen based on the *RBD* of block-2 in Fig 2.8 and consequent is defined the same as in block-1. Total $9 = 3^2$ rules shown in Table 2.3 are constructed in this scheme.

- Antecedent 1 - The remaining battery level of node x.
- Antecedent 2 - The remaining battery level of node y.

Table 2.2. Complete 27 rules for square-grid block-1

Rule Number	Antecedent 1	Antecedent 2	Antecedent 3	Consequent
1	Low	Low	Low	Very Low
2	Moderate	Low	Low	Very Low
3	High	Low	Low	Low
4	Low	Moderate	Low	Low
5	Moderate	Moderate	Low	Moderate
6	High	Moderate	Low	High
7	Low	High	Low	Moderate
8	Moderate	High	Low	High
9	High	High	Low	Very High
10	Low	Low	Moderate	Very Low
11	Moderate	Low	Moderate	Low
12	High	Low	Moderate	Moderate
13	Low	Moderate	Moderate	Moderate
14	Moderate	Moderate	Moderate	High
15	High	Moderate	Moderate	Very High
16	Low	High	Moderate	Moderate
17	Moderate	High	Moderate	High
18	High	High	Moderate	Very High
19	Low	Low	High	Low
20	Moderate	Low	High	Moderate
21	High	Low	High	Moderate
22	Low	Moderate	High	High
23	Moderate	Moderate	High	Very High
24	High	Moderate	High	Very High
25	Low	High	High	High
26	Moderate	High	High	Very High
27	High	High	High	Very High

2.4.3.2 Hex-Grid Sensor Network

The analysis for the hex-grid is carried out on the same lines as that of the square-grid. Fig 2.2 (b) shows that as in the case of a square grid, two neighboring node failures cause network failure. The RBD block of a single node is shown in Fig. 2.9.

Table 2.3. Complete 9 rules for square-grid block-2

Rule Number	Antecedent 1	Antecedent 2	Consequent
1	Low	Low	Very Low
2	Moderate	Low	Low
3	High	Low	Moderate
4	Low	Moderate	Low
5	Moderate	Moderate	High
6	High	Moderate	Very High
7	Low	High	Moderate
8	Moderate	High	Very High
9	High	High	Very High

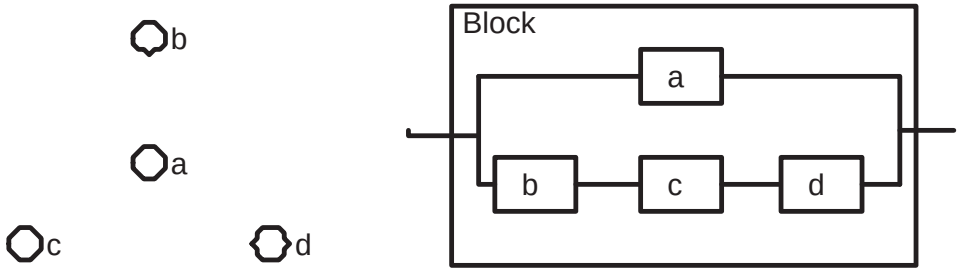


Figure 2.9. RBD block for a single node in the Hex-grid.

Since the relation between a node and all of its neighbors is modeled by its corresponding *RBD* block, the *RBD* block's for the neighbors is not constructed as this causes the relationship between the nodes to be considered twice. In the hex-grid network, things simplify a lot when we abstract two antecedents from its hexagonal structure. The two antecedents are listed below:

- Antecedent 1 - The remaining battery level of node a.
- Antecedent 2 - The minimum remaining battery level of node b, c and d.

Consequent definition, levels to represent antecedent and consequent are consistent with Section 2.4.1. Total $9 = 3^2$ rules constructed for hex-grid network are listed in Table 2.3.

Table 2.4. Complete 9 rules for Hex-grid network

Rule Number	Antecedent 1	Antecedent 2	Consequent
1	Low	Low	Very Low
2	Moderate	Low	Low
3	High	Low	Moderate
4	Low	Moderate	Low
5	Moderate	Moderate	High
6	High	Moderate	Very High
7	Low	High	Moderate
8	Moderate	High	Very High
9	High	High	Very High

2.5 Simulation and Discussion

In the previous sections, it is theoretically addressed the approach to analyze wireless sensor network lifetime using interval type-2 fuzzy logic systems. Research on the single node lifetime justified the use of Gaussian MFs with uncertain standard deviation (std) to model node lifetime and type-2 fuzzy logic systems for the network lifetime likewise. We are now ready to validate the feasibility of our approach. In this section, we will test our new Type-2 Fuzzy Logic Lifetime Evaluator(FLLE2) and train the design parameters of FLLE2 with testing data. A singleton type-1 fuzzy logic system is devised to compare the performance with FLLE2.

2.5.1 Energy Consumption Model and Test Data Generation

Simulation is implemented on two deployment modes: square-grid and hex-grid. For both modes, the basic units in Fig 2.8 block-1 and Fig 2.9 are considered and deploy the network with minimum density (referring to Section 2.1.1). All sensor nodes are initialized with power level up to $10J(Joule)$. Note our simulation does not start from the very beginning of deployment. Our explicit goal is to evaluate the remanent active time of a network when some sensor nodes have consumed certain amount of energy in

task performing. Thus we initialize the remaining power level of all sensor nodes by random variables within $[0, 10]J$.

Radio energy consumption model has been well-studied in LEACH (Energy-Efficient Communication Protocols for Wireless Microsensor Networks) [48] and ESO (Energy-Efficient Self-Organization for Wireless Sensor Networks) [49]. Different applications consume energy resources in different manners. According to the path loss model in wireless communication, the energy E needed to transmit over distance d is proportionate to d^β give by $E \sim d^\beta$, where β is the path loss exponent depending on the specific propagation environment. For example, β will have a larger value for long distance transmission than for short distance transmission. In order to save energy resources and decrease interference, power control is widely used in wireless communications such that the radio could be adjusted for a certain range of output power level. The following model is adopted from [48] where perfect power control is assumed.

To transmit l bits over distance d, the sender's radio spends

$$E_{TX}(l,d) = E_{TX-elec}(l) + E_{TX-amp}(l,d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2 & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4 & d \geq d_0 \end{cases} \tag{2.9}$$

and to receive this message, the receiver's radio spends

$$E_{RX}(l,d) = E_{RX-elec} = lE_{elec}. \tag{2.10}$$

The electronics energy, E_{elec}, depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\epsilon_{fs}d^2$ in free-space or $\epsilon_{mp}d^4$ in multi-path environment, depends on the distance to the receiver and the acceptable bit-error rate. For the simulations described in later sections, the commu-

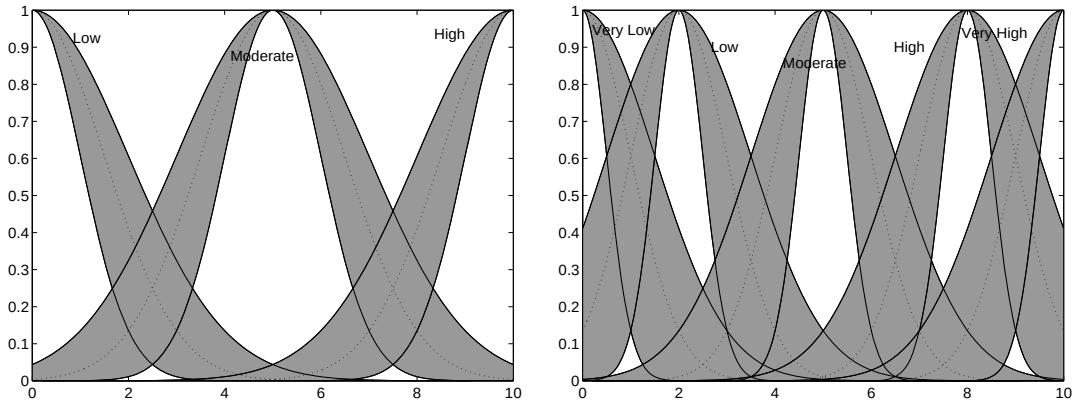
nication energy parameters are set as: d_0=86.2m, the radio dissipates E_{elec}=50nJ per bit to run the transmitter or receiver circuitry, ϵ_{fs}=10$pJ/bit/m^2$ and ϵ_{mp}=0.0013$pJ/bit/m^4$.

In the simulation, the formula in (2.9) under the circumstances $d < d_0$ is used. Sensor nodes take turns to transmit data to their immediate neighbors by broadcast until network failure. It is assumed the transmission alternation is determined by the remaining power level of individual sensor nodes at the beginning of each epoch. Sensor node with the lowest power level gets to transmit with the least probability. We also make the assumption that the transmitted data bits per turn is a constant such that the energy dissipation varies only with the transmitting distance. Simulations are run for square-grid and hex-grid wireless sensor networks using the OPNET platform set up in [49]. The actual sensor network lifetime can be obtained for different sensor battery level $[0, 10]$J settings.

Data sets of $N = 600$ are collected for square-grid and hex-grid networks respectively. Each data set consists of battery levels of 4 nodes and actual network lifetime. In the FLS network lifetime analysis, 300 data sets are used for training and the remaining 300 data sets were used for testing.

2.5.2 Simulation Results and Discussion

A type-2 Fuzzy Logic Lifetime Evaluator (FLLE2) and a type-1 Fuzzy Logic Lifetime Evaluator (FLLE1) are designed for a square-grid wireless sensor network. The initial 27 rules were designed according to Table 2.2. The antecedent and consequent MFs for FLLE1 and FLLE2 are plotted in Fig. 2.10. Then the parameters of FLLE1 and FLLE2 are tuned using steepest-descent algorithm. We followed the training algorithm proposed in [42] for FLLE2. 300 data sets were used for training. Both FLLE1 and FLLE2 were trained for 6 epochs. After training, the parameters of FLLE1 and FLLE2 were fixed and the remaining 300 data sets were used for testing.



(a) (b)

Figure 2.10. (a)Antecedent MFs (b)Consequent MFs.

Fig. 2.11 summarized the root-mean-square-errors (RMSE) between the estimated lifetime and the actual lifetime. Observe that the RMSEs of both FLLE1 and FLLE2 for the square-grid wireless sensor network decrease along with the tuning epoch and the FLLE2 performs much better than the FLLE1. For example, FLLE1 takes down RMSE from the initial 0.1472 to around 0.037 at the $6th$ epoch while FLLE2 reduces RMSE from 0.104 to 0.006. FLLE2 achieves nearly 83.8% reduction in RMSE comparing to FLLE1 at the $6th$ epoch.

Similarly, a FLLE1 and a FLLE2 are designed for hex-grid wireless sensor network. The initial 9 rules were set according to Table 2.4. The same antecedent and consequent MFs as in square-grid network were used. The FLLE1 and FLLE2 are then trained and tested. Fig. 2.12 plotted the RMSE between the estimated lifetime and the actual lifetime. Results show that the FLLE1 and the FLLE2 for a hex-grid sensor network decrease the RMSE along with the tuning epoch, and the FLLE2 outperforms the FLLE1 likewise. For example, FLLE1 reduces the RMSE from the initial 0.1734 to around 0.069 at the $6th$ epoch while the FLLE2 reduces the RMSE from 0.1038 to 0.006. The FLLE2 achieves nearly 91.3% reduction in RMSE comparing to FLLE1 at the $6th$ epoch.

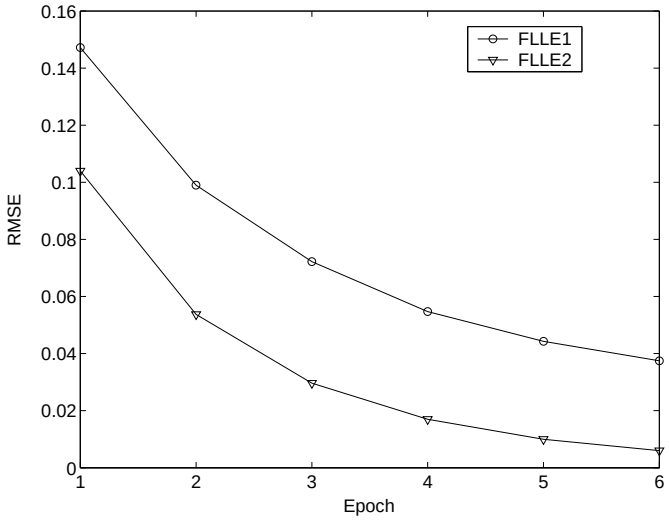


Figure 2.11. Square-Grid: RMSE of FLLE1 and FLLE2.

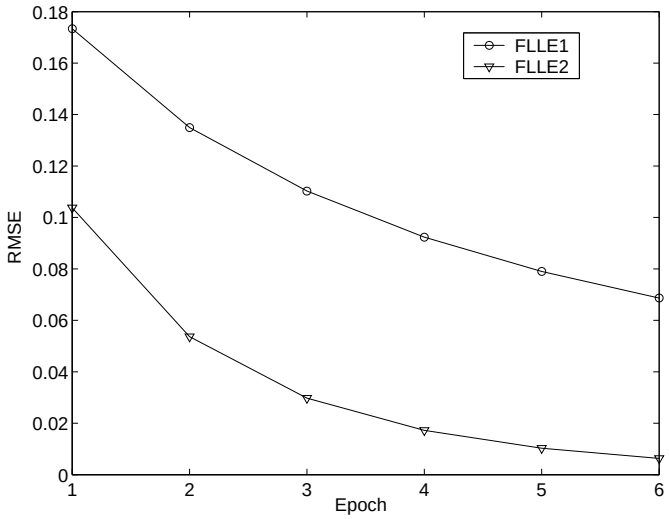


Figure 2.12. Hex-Grid: RMSE of FLLE1 and FLLE2.

Up to the present, the basic units of square-grid and hex-grid wireless sensor networks are discussed. Now, let N_{min} be the number of sensor nodes required to be deployed with minimum density (referring to Section 2.1.1). For square-grid sensor network, the network *RBD* in Fig. 2.8 consists of $(\sqrt{N_{min}}-1)^2$ block-1's and $2(\sqrt{N_{min}}-1)$ block-2's in series, the whole square grid network can actually be decomposed into multiple blocks

serial connected together and the method of setting up rules can be applied likewise. Similarly, in hex-grid sensor network, $N_{min}/2$ RBD blocks in Fig. 2.9 connected in series represent the network which can be decomposed the same way as in the square-grid sensor network.

CHAPTER 3

DISTRIBUTED SENSOR NETWORKS DEPLOYMENT

3.1 Overview of Type-1 Fuzzy Logic Systems

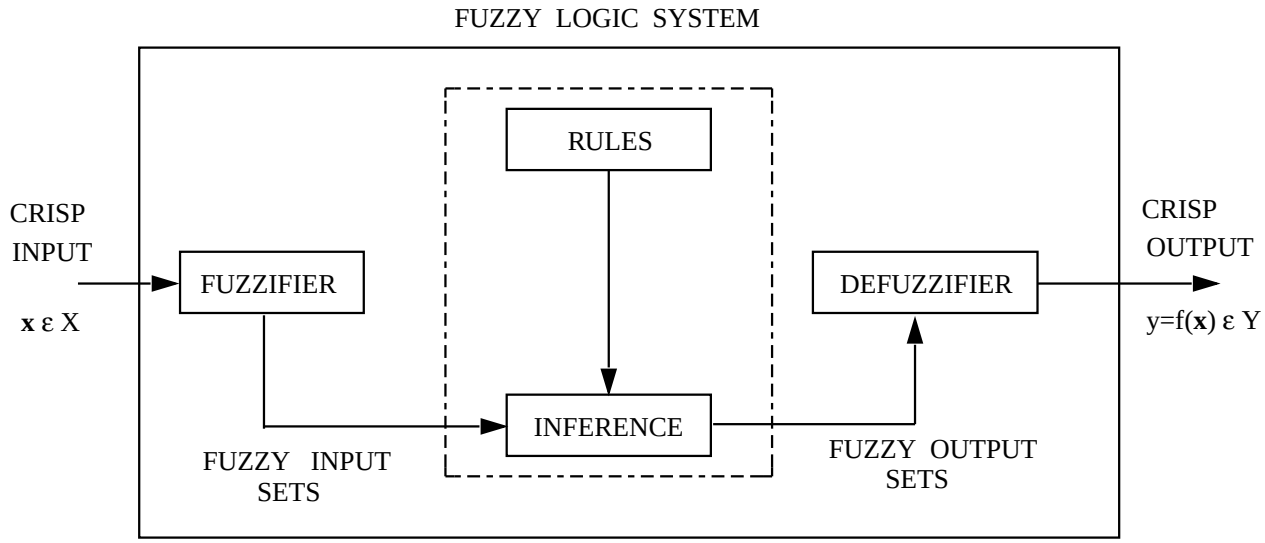


Figure 3.1. The structure of a fuzzy logic system.

Figure 3.1 shows the structure of a rule-based type-1 fuzzy logic system (FLS) [43]. It contains four components: fuzzifier, rules, inference engine and defuzzifier. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets.

Rules are the heart of a FLS and may be provided by experts or can be extracted from numerical data. In either case, the rules that we are interested in can be expressed as a collection of IF-THEN statements, e.g. [50],

IF the total average input rate of real-time voice and video traffic is a *moderate amount*, and the total average input rate of the non-real-time data traffic is *some*,

THEN the confidence of accepting the telephone call is a *large amount*.

The IF-part of a rule is its *antecedent* and the THEN-part of a rule is its *consequent*.

The process of making a crisp input fuzzy is called *fuzzification*. The most widely used fuzzification is the *singleton fuzzification*. All fuzziness for a particular fuzzy set is essentially characterized by the *membership functions*. The shapes used to describe the fuzziness have very few restrictions, but with the help of mathematical structure, some standard terms related to the shape of membership functions have been developed over the years [51]. The most common forms of membership functions are those that are normal and convex.

Consider a type-1 FLS having p inputs and one output. Suppose that it has M rules, where the lth rule has the form:

R^l : IF x_1 is F_1^l and x_2 is F_2^l and $\cdots$ and x_p is F_p^l, THEN y is G^l. $l = 1, ..., M$

Assuming singleton fuzzification is used, when an input $\mathbf{x}' = \{x'_1, \ldots, x'_p\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{\mathrm{F}_1^l}(x'_1) \star \mu_{\mathrm{F}_2^l}(x'_2) \star \cdots \star \mu_{\mathrm{F}_p^l}(x'_p) = \mathcal{T}_{i=1}^{p} \mu_{\mathrm{F}_i^l}(x'_i) \tag{3.1}$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm.

The last but not the least process in a FLS is called *defuzzification*. Defuzzification is the conversion of fuzzy output sets to crisp output sets. There are many defuzzification methods including maximum, mean-of-maxima, centroid, center-of-sums, height, modified height and center-of-sets. This paper focuses for illustrative purposes, on the center-of-sets defuzzifier [50]. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average

of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p}\mu_{\mathrm{F}_i^l}(x_i')$, so that

$$y_{cos}(\mathbf{x}') = \frac{\sum_{l=1}^{M} c_{\mathrm{G}^l}\mathcal{T}_{i=1}^{p}\mu_{\mathrm{F}_i^l}(x_i')}{\sum_{l=1}^{M}\mathcal{T}_{i=1}^{p}\mu_{\mathrm{F}_i^l}(x_i')} \tag{3.2}$$

where M is the number of rules in the FLS.

The next section will detail the design of the rule-based type-1 FLSs for distributed sensor deployment issue.

3.2 Design of FLSs for Distributed Sensor Deployment

3.2.1 Assumptions and Notations

In this research, several assumptions are made:

- Sensor field is denoted by a two-dimensional grid. Sensing and communication is modeled as a circle on this grid.
- Coverage discussed in this paper is grid coverage. A grid point is covered when at least one sensor covers this point.
- A sensor can detect or sense any event within its sensing range, denoted by Rs. Coverage is determined based on Rs.
- Two sensors within their communication range, denoted by Rc can communicate with each other. Neighbors of a sensor are defined as nodes within its communication range.
- All sensor nodes are assumed peer to peer.
- Sensor nodes have certain mobility and are capable of computing, detection and communication.
- Sensor node can obtain the knowledge of its location.
- Sensors are synchronized by coherence time. A one-time move is made within each coherence period.

3.2.2 FLSs Design for Distributed Sensor Deployment

Fuzzy logic system is well known to be able to handle uncertainty and ambiguity. In practice not all uncertainty is random. Some forms of uncertainty are non-random and hence not suited to treatment or modeling by probability theory. Fuzzy set theory is a marvelous tool for modeling uncertainty associated with vagueness, or with a lack of information regarding a particular element of the problem at hand. Concerning the distributed sensor deployment, the moving distance and direction of each sensor are distributed and full of uncertainty which can barely be described by some random distribution. A fuzzy logic system is well known as model free. Their membership functions are not based on statistical distributions. Therefore we propose to apply a fuzzy logic system to the distributed sensor deployment problem. Each sensor makes a fully distributed decision on its movement based on FLS.

The algorithm starts with random deployment. Assume a two-dimensional sensor field is the target area of surveillance. In the initial condition, a given number of sensors are randomly deployed. Because of the randomness in initial deployment, it is very likely the sensor field will not be fully covered. Part of the sensor field might be over crowded with sensors. Such unbalanced deployment brings difficulty in target detection and tracking, and increases the interference during communications. Fig. 3.2 gives an example of a randomly deployed field. As shown in Fig. 3.2, targets in the uncovered area cannot be detected while in the over crowded area, communication between sensors is corrupted by the interference from neighboring nodes.

The algorithm then intends to re-deploy the sensors such that maximum field coverage and high quality communication could be achieved. Each individual sensor in the network needs to fine-tune its location such that densely deployed sensors can be evenly spreaded in the field. Two critical procedures are considered in the algorithm:

- Determine the next-step move distance for each sensor.

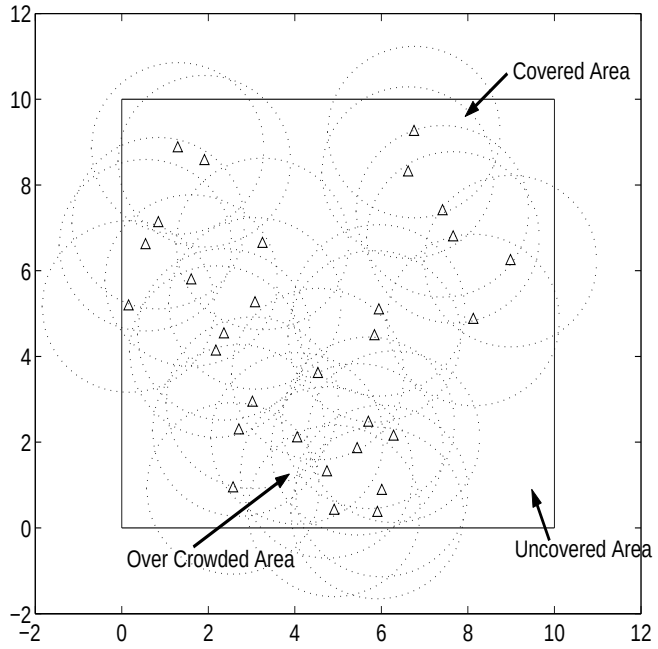


Figure 3.2. An Example of Random Deployment.

- Determine the next-step move direction for each sensor.

The next-step move distance is hard to determine. If the move distance is too small or too large, each step consumes more time and energy to get a stable deployment. Excessive move and oscillation is unavoidable in previous work with no fuzzy system. This paper designs a fuzzy logic system to determine the next-step move distance for each sensor.

An ideal sensor deployment will have uniform distribution for better coverage. But in random deployment, coverage uniformity is hard to achieve initially. In a sensor network composed of mobile sensors, each sensor detects the number and location of its neighbors and decides its *neighborhood density*. If the sensor has a high density of neighboring nodes, it makes decision using FLSs to shift a certain distance away from the high density area. If the neighborhood density is low, the sensor might stand still or shift a little distance away from the current location.

As illustrated in Fig. 3.2, the *neighborhood density* of a sensor node is determined by two factors: the number of neighbors and the distance between a sensor node and its neighbors. Based on this knowledge, two antecedents are chosen as follows:

Antecedent 1. Number of neighbors of each sensor.

Antecedent 2. Average Euclidean distance between sensor node and its neighbors

The linguistic variables to represent the number of neighbors for each sensor are divided into three levels: *high*, *moderate* and *low*; and those to represent the average the Euclidean distance between sensor node and its neighbors are divided into three levels: *far*, *moderate* and *near*. The consequent - the shift distance normalized by sensing range Rs is divided into three levels: *far*, *moderate* and *near*. Table 1 summaries the rules and consequents.

Table 3.1. Fuzzy Rules and Consequent

Antecedent 1	Antecedent 2	Consequent
Low	Near	Moderate
Low	Moderate	Near
Low	Far	Near
Moderate	Near	Far
Moderate	Moderate	Moderate
Moderate	Far	Near
High	Near	Far
High	Moderate	Moderate
High	Far	Moderate

One example of rules is as follows:

IF the number of neighbors of sensor i is *high* and average Euclidean distance between sensor i and its neighbors is *moderate*, THEN the normalized shift distance of sensor i should be *high*.

9 rules are set up for this FLS because every antecedent has 3 fuzzy sub-sets and there are 2 antecedents. Trapezoidal membership functions (MFs) are used to represent *high*, *low*, *far* and *near* and triangle MFs to represent *moderate*. Two antecedents are normalized to the range $[0, 10]$. These membership functions are shown in Fig. 3.3.

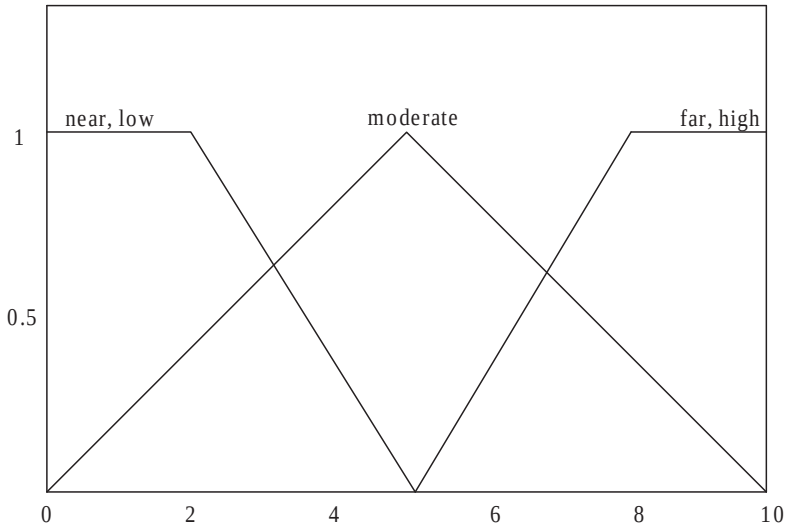


Figure 3.3. Antecedent Membership Functions.

Applying center-of-sets defuzzification [7], for every input (x_1, x_2), the output is computed using

$$y_{(x_1,x_2)} = \frac{\sum_{l=1}^{9} c_{\mathrm{G}^l} \mu_{\mathrm{F}_1^l}(x_1) \mu_{\mathrm{F}_2^l}(x_2)}{\sum_{l=1}^{9} \mu_{\mathrm{F}_1^l}(x_1) \mu_{\mathrm{F}_2^l}(x_2)} \tag{3.3}$$

Repeating these calculations for $\forall x_i \in [0, 10]$,we obtain a decision surface $y(x_1, x_2)$ as shown in Fig. 3.4.

Generally, the decision surface is time-varying and nonlinear. From Fig. 3.4, although the number of neighbors for a particular sensor is high, the move distance can be smaller than some sensor with fewer "crowded" neighbors, i.e. very close average

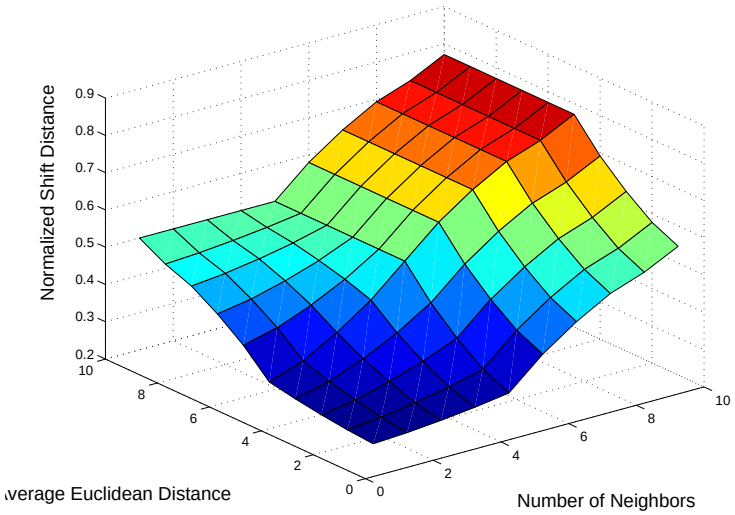


Figure 3.4. Control Surface of Shift Distance.

Euclidean distance between the sensor and its neighbors. With the assist of the decision surface, the next-step move distance can be determined.

Comparing to move distance, the next-step move direction is much easier to decide. Coulomb's law in physics becomes a useful tool to tackle the problem. For instance, assume sensor i has 2 neighbors in its communication range as shown in Fig. 3.5.

The coordinate of sensor i is denoted as $C_i = (X_i, Y_i)$.

The next-step move direction of sensor i could be represented as follows:

$$\vec{v} = \sum_{j=1}^{2} \frac{\vec{C_j} - \vec{C_i}}{|\vec{C_j} - \vec{C_i}|^2} \tag{3.4}$$

$$tan(\alpha) = \frac{Y_{(\vec{v})}}{X_{(\vec{v})}} \tag{3.5}$$

After getting distance and direction (angle α) , sensor i clearly knows the next-step move information. In order to prolong the battery life of each individual sensor, a coherence time is introduced as the duty cycle during which the changes of two antecedents

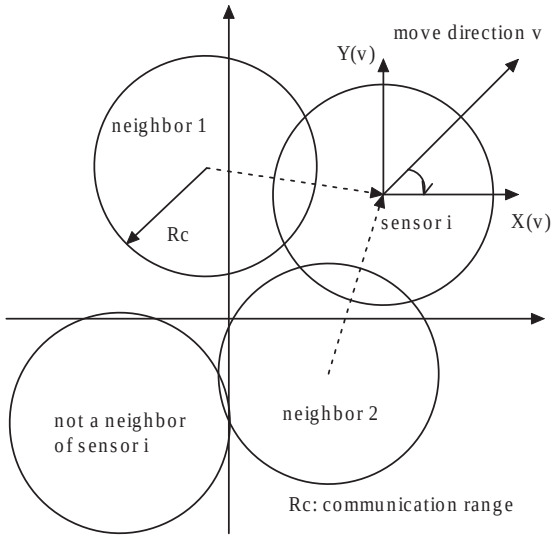


Figure 3.5. Example of Next Step Move Direction.

can be ignored. Sensors are put into idle or sleep mode if within the coherence time, the information of neighbors remains unchanged.

3.3 Simulation and Discussion

Simulation investigates various number of sensors deployed in a field of 10×10 square kilometers area. We assume each sensor is equipped with an omni-directional antenna to carry out the task of detection and communication. Evaluation of our scheme follows three criteria: field coverage, converging speed, mean travel distance per node and outage probability. Results are averaged over 200 Monte Carlo simulations.

The performance of the proposed algorithm is compared with the Distributed Self Spreading Algorithm (DSSA) proposed in [20]. DSSA is known as a good solution in the self-deployment of mobile sensor nodes. The main idea of DSSA is to define a partial force for the movement of sensors during the deployment process. The force a node receives from a closer neighbor node is greater than that from a farther neighbor. For N

sensor nodes deployed in a square field with area A, DSSA formulates the partial force sensor node i receives from neighbor node j as:

$$f_n^{i,j} = \frac{D}{\mu^2}(R_c - \left|p_n^i - p_n^j\right|)\frac{p_n^i - p_n^j}{\left|p_n^i - p_n^j\right|} \tag{3.6}$$

where R_c stands for communication range, $\mu = \frac{N \cdot \pi \cdot R_c^2}{A}$ is called the expected density while D is the local density, and p_n^i stands for the location of node i at time step n. Each node makes decision to move by adding up all partial forces from its neighboring nodes. DSSA sets up two criteria: stable status limit (S_{lim}) and oscillation limit (O_{lim}) to stop a sensor's movement.

Fig. 3.6 shows at 2 kilometer sensing range (R_s=2km) and 4 kilometer communication range (R_c=4km), the coverage of the initial random deployment, the coverage after DSSA is implemented and the coverage after using FLSs. 3 iterations are run for all three schemes. When 20 sensors are deployed, the coverage after random deployment was initially around 85% and the DSSA increased it to 93%. After FLSs were used, the coverage reached approximate 98% after 3 iterations.

Fig. 3.7 gives the results when 10 iterations are completed for the three deployment schemes. We observe that the performance of DSSA gets closer to our FLSs after 10 iterations. Both FLSs and DSSA can dramatically increase the network coverage in the low density network. Fig. 3.6 and Fig. 3.7 also indicates that instead of deploying large amount of sensors, the desired field coverage could be achieved with fewer sensors. Comparing Fig. 3.6 and Fig. 3.7, the FLSs increases the network coverage faster than the DSSA in terms of iteration times.

Fig. 3.8, Fig. 3.9 and Fig. 3.10 are the real pictures of 20 sensors from random deployment, after implementing FLS and DSSA respectively. Both FLS and DSSA can

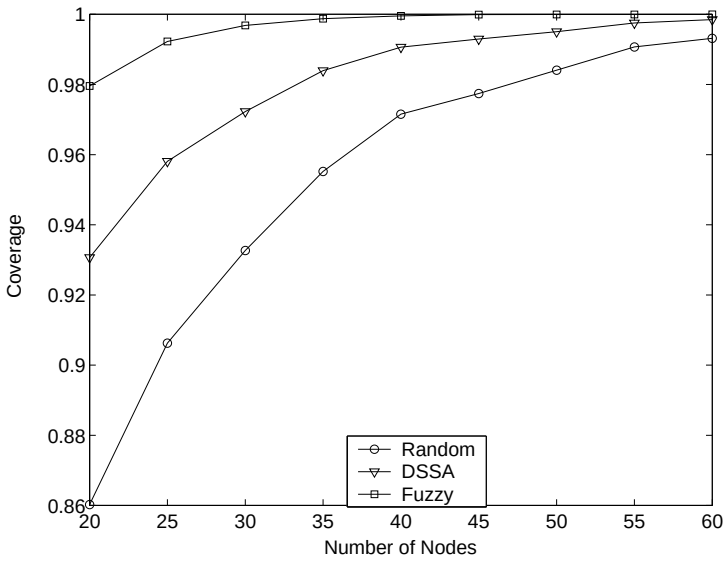


Figure 3.6. Coverage vs. Number of Sensors, 3 iterations.

spread the densely deployed sensors but the deployment after using FLS demonstrates more uniformity than the one using DSSA.

Next two cases are simulated when 30 sensor nodes and 60 sensor nodes are deployed respectively. Network coverage according to these two cases are presented in Fig. 3.11 and Fig. 3.12.

It is fairly clear in Fig. 3.11 and Fig. 3.12 that the FLSs increase the network coverage much faster than the DSSA. For instance, when 30 sensor nodes were deployed, the FLSs boost the network coverage from the initial 93% to around 98.5% in only 1 iteration whereas the DSSA takes 6 iterations to reach the same coverage.

The average distance traveled by each sensor node is also important in the energy saving problem. For energy constrained wireless sensor nodes, less travel distance leads to less energy consumption. The goal is to adjust sensors' positions appropriately such that the maximum coverage is achieved with minimum energy dissipation in deployment. We calculated the average distance traveled by each sensor node for the FLS and compared it against the DSSA since both reach the same network coverage. Results in Fig. 3.13

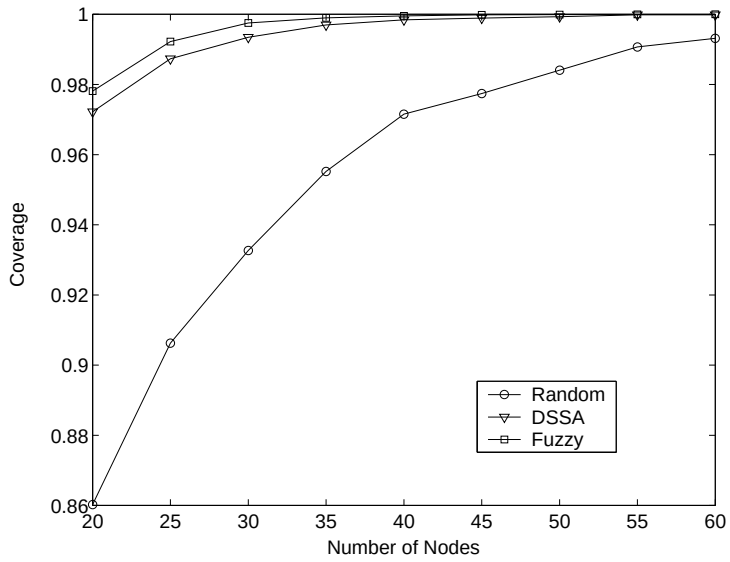


Figure 3.7. Coverage vs. Number of Sensors, 10 iterations.

indicate that for the FLS scheme, each sensor node travels less average distance than that in the DSSA scheme. Furthermore, in FLS scheme, the average travel distance by each node varies little when the number of sensors changes, which implies that the energy consumed in deployment is nearly independent of network density.

In wireless sensor networks, the radio link performance is usually limited by interference rather than noise, therefore, the probability of outage due to co-channel interference is of primary concern. Measurements [52] have shown that at any value of $d_{i,j}$ (the Euclidean distance between sensor i and sensor j), the path loss $PL(d_{i,j})$ is random and distributed log-normally (normal in dB) about the mean distance dependent value. That is:

$$PL(d_{i,j})[dB] = \overline{PL}(d_{i,j}) + X_\sigma = \overline{PL}(d_0) + 10nlog(\frac{d_{i,j}}{d_0}) + X_\sigma \tag{3.7}$$

and

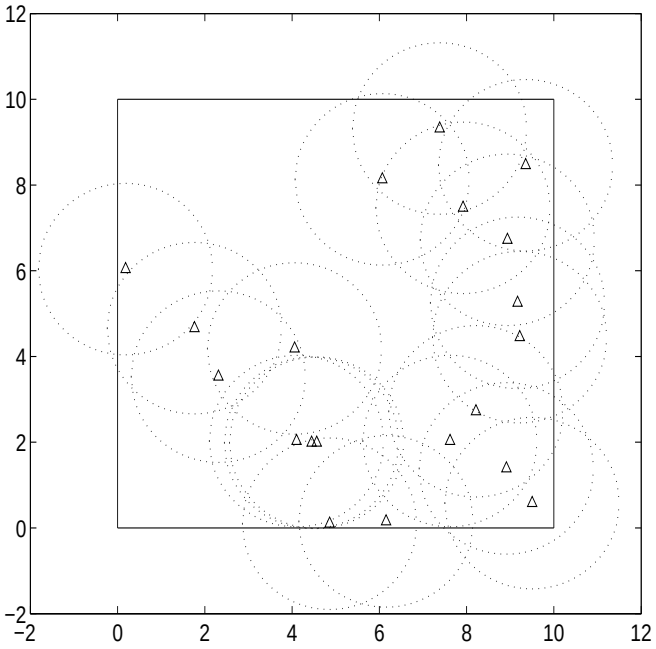


Figure 3.8. Random deployment with 20 sensors. (Rc=4,Rs=2, 10*10 area).

$$P_r(d_{i,j})[dBm] = P_t[dBm] - PL(d_{i,j})[dB] \tag{3.8}$$

where X_σ is a zero-mean Gaussian distribution random variable (in dB) with standard deviation σ (also in dB).

The log-normal distribution describes the random *shadowing* effects on the propagation path which implies that measured signal levels at a certain distance have a Gaussian (normal) distribution about the distance-dependent mean and standard deviation σ. Since $PL(d_{i,j})$ follows normal distribution, so is $P_r(d_{i,j})$, and the Q function may be used to determine the probability that the received signal level will exceed (or fall below) a particular level.

The probability that the received signal level will exceed a certain value γ can be calculated from the cumulative density function as

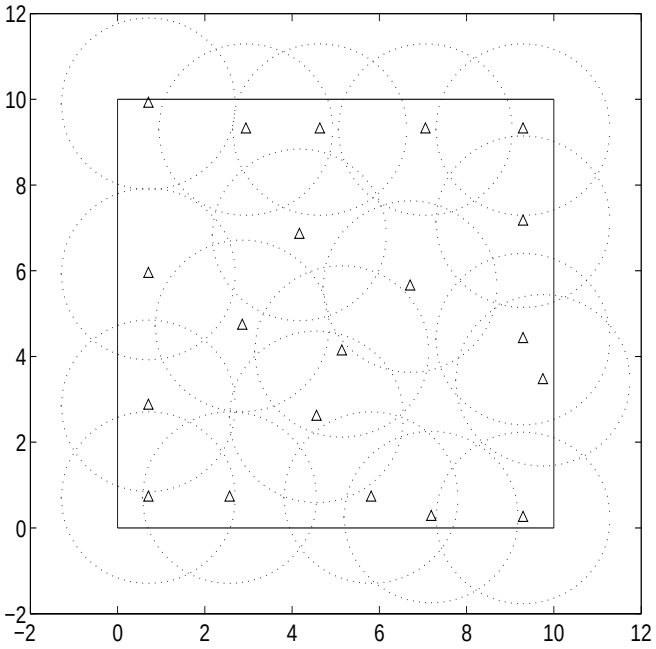


Figure 3.9. Deployment with 20 sensors after implementing FLS.

$$P_r[P_r(d_{i,j}) > \gamma] = Q\left(\frac{\gamma - \overline{P_r}(d_{i,j})}{\sigma}\right) \tag{3.9}$$

For sensor i with N neighbors, if the sensor i acts as the destination node during one communication message, the signal to interference ratio (SIR) is represented as:

$$SIR(i) = \frac{P_r(d_{i,j})}{\sum_{k=1}^{N} P_r(d_{i,k})}, k \neq j \tag{3.10}$$

The denominator denoting the effect of co-channel interference is a sum of $N-1$ log-normal signals. Evaluating the outage probability requires the probability distribution of the interference power. There is no known exact expression for the probability distribution for the sum of log-normal random variables, but various authors have derived several approaches which approximate the sum of log-normal random variables by another log-normal random variable.

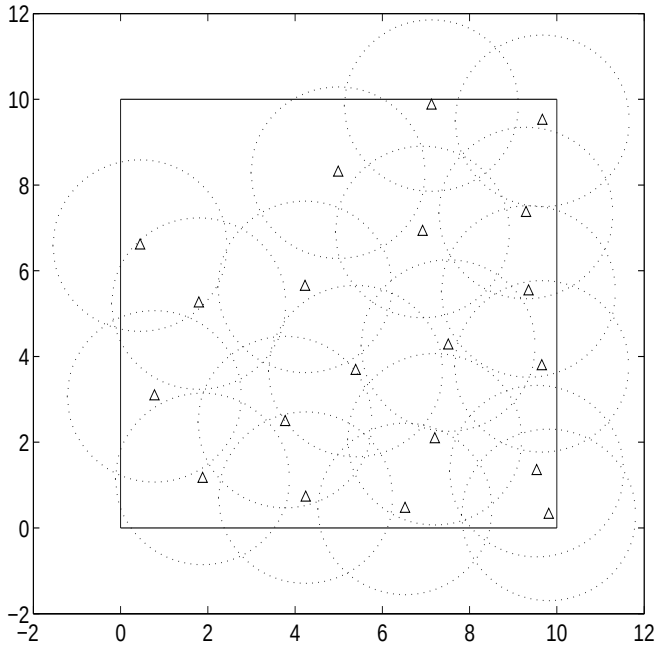


Figure 3.10. Deployment with 20 sensors after implementing DSSA.

This paper used Fenton-Wilkinson method [53]. The co-channel interference can now be approximated by one log-normal random variable. SIR(in dB) as a result follows log-normal distribution as well. We expatiate the Fenton-Wilkinson method in the Appendix. Results of outage probability are presented in Fig.3.14.

Observe in Fig. 3.14 that the FLSs scheme successfully reduced the outage probability by nearly 15% compared to the DSSA method when the number of sensors is 60. This implies a higher probability that the received signal level will exceed the SIR threshold using the FLSs scheme. The DSSA did not perform well in the outage probability because it did not take the outage probability into performance evaluation [20].

It is introduced earlier that DSSA stops a sensor's movement by two criteria: stable status limit (S_{lim}) and oscillation limit (O_{lim}). Reference [20] shows that it takes more than 10 times the iterations to termination. Then the fuzzy approach gains a distinct advantage over DSSA by converging in around 3 iterations. Thus a stop criteria is not

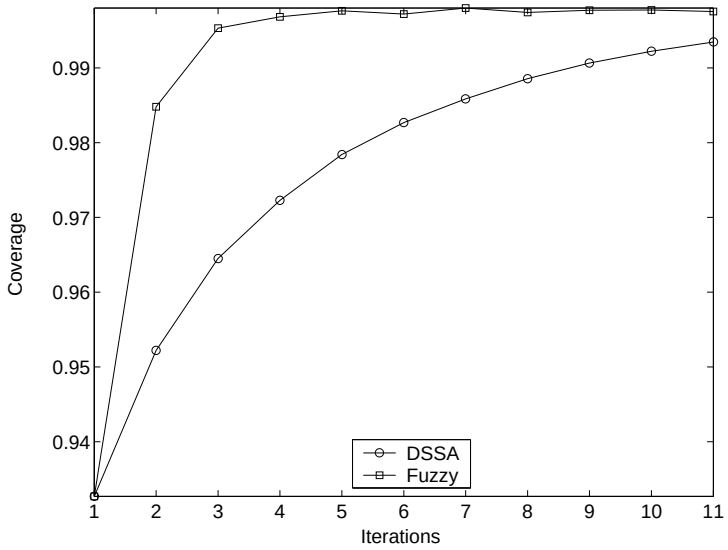


Figure 3.11. Coverage vs. Number of Iterations, 30 Nodes.

required in this fuzzy approach. These facts indicate that this FLS scheme is much faster and simpler to implement comparing to the DSSA method and more significantly, the FLS scheme maximizes the network coverage with less energy consumption in deployment.

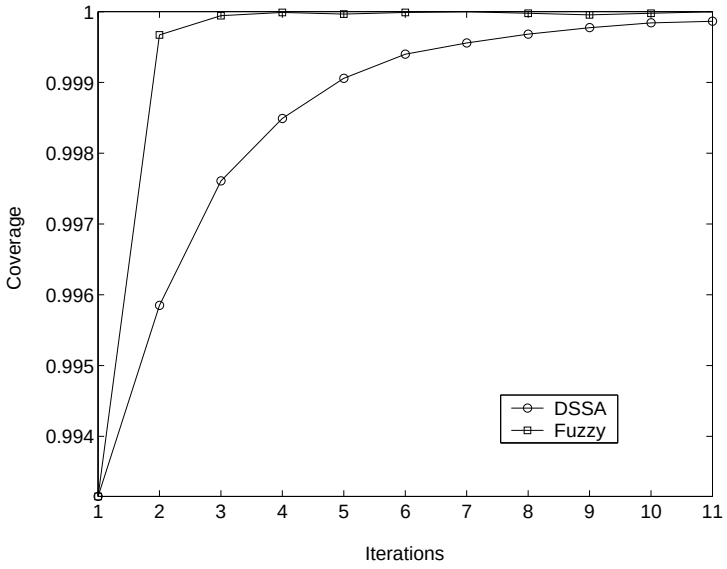


Figure 3.12. Coverage vs. Number of Iterations, 60 Nodes.

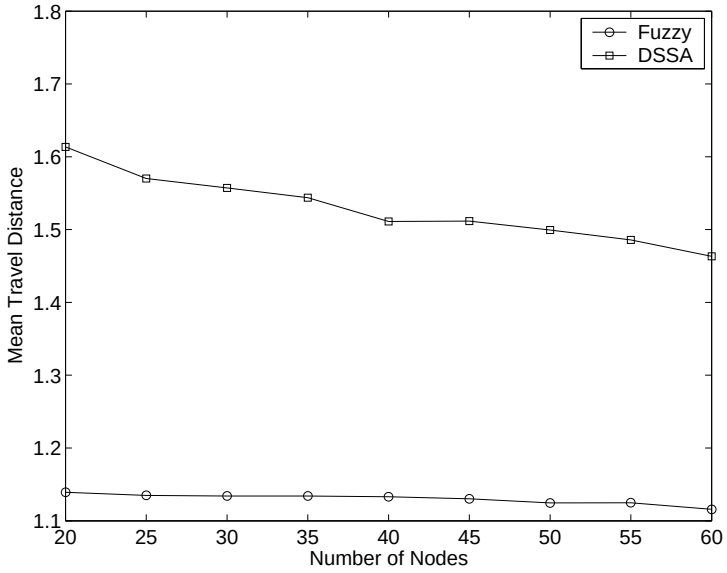


Figure 3.13. Travel Distance vs. Number of Sensors Deployed.

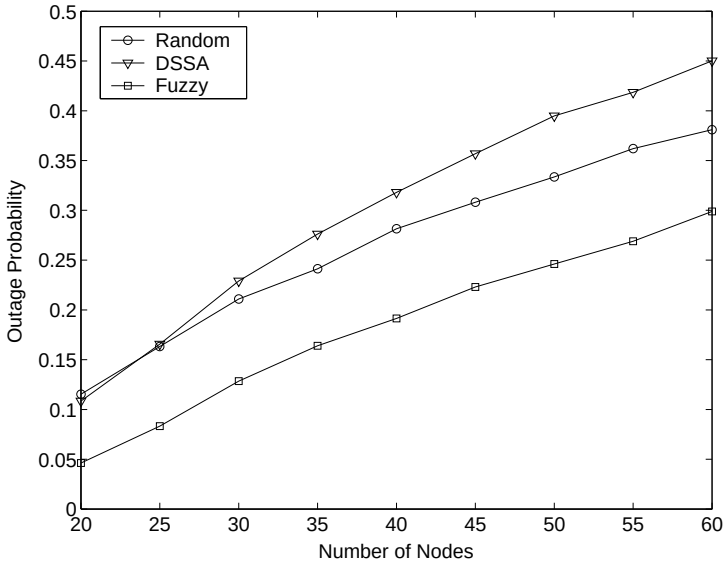


Figure 3.14. Outage Probability vs. Number of Sensors (Rc=4,Rs=2).

CHAPTER 4

SPECTRUM EFFICIENT CODING SCHEME

4.1 Prelimiaries

This section reviews the basic concepts of distributed source coding for correlated information and introduce Slepian-Wolf coding for lossless source coding and Wyner-Ziv coding for the lossy case.

Consider a distributed wireless sensor network consisting of individual sensors that monitor the sensor field. These sensors transmit their highly correlated data to a central hub node to reconstruct the observations. Transmission of redundant information can be easily avoided if the sensors communicate with each other but such inter-node cooperation requires higher bandwidth and consumes more energy in communication. Slepian and Wolf in [54] proved that if there is no communication among the sensors, theoretically there was no loss in performance under certain conditions. The Slepian-Wolf theorem has been extended to the lossy coding of continuous-valued sources by Wyner and Ziv [54,55].

4.1.1 Slepian-Wolf Coding

Let X and Y be two correlated independent and identically distributed ($i.i.d$) binary sources. For lossless compression with $X' = X$ and $Y' = Y$ after decompression, from Shannon's source coding theory [56], a rate given by the joint entropy $H(X, Y)$ of X and Y is sufficient if encoding them together.

Fig. 4.1 gives an example of joint encoding and distributed encoding of two binary sources. In Fig. 4.1 (a), encoder X compress X into $H(X)$ bits per sample and based on the complete knowledge of X at both encoder and decoder, Y is then compressed into

$H(Y|X)$ bits per sample, while in Fig. 4.1 (b), encoder X and Y do not communicate and perform separate encoding.

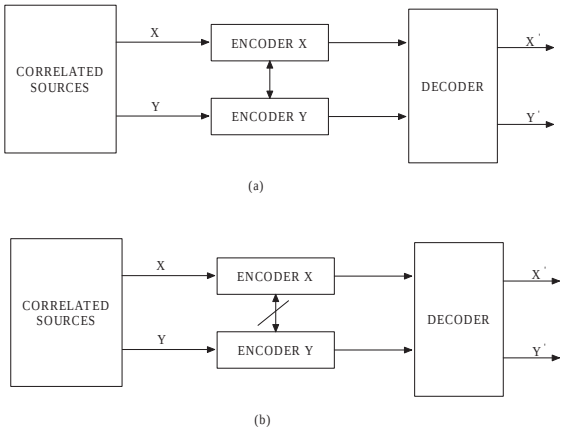


Figure 4.1. (a) Joint encoding. (b) Distributed encoding.

The Slepian-Wolf theorem [54] states that if X and Y are correlated according to some arbitrary probability distribution $p(x, y)$, then X can be compressed separately (without access to Y) without losing performance compared to the condition in Fig. 4.1 (a). It says that the achievable region of DSC for discrete sources X and Y is given by $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ and $R_X + R_Y \geq H(X, Y)$, which is shown in Fig. 4.2.

For practical Slepian-Wolf coding, the first attemp is to approach the corner point A in the Slepian-Wolf rate region of Fig. 4.2 with $R_1 + R_2 = H(X|Y) + H(Y) = H(X, Y)$. This is actually a problem of source coding of X with side information Y at the decoder as shown in Fig. 4.3. Similarly the other corner point B of the Slepian-Wolf rate region can be approached by exchanging the roles of X and Y and all points between the two corner points can be realized by time-sharing.

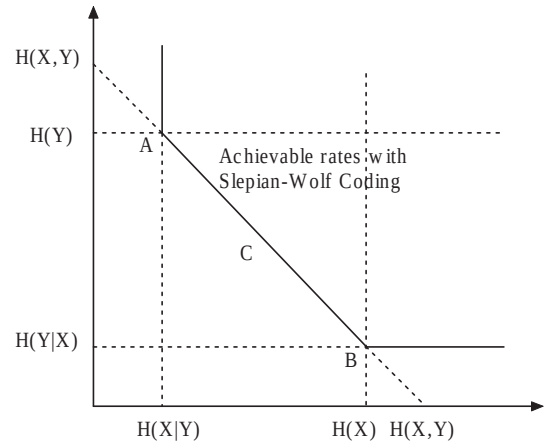


Figure 4.2. The Slepian-Wolf region for two binary sources.

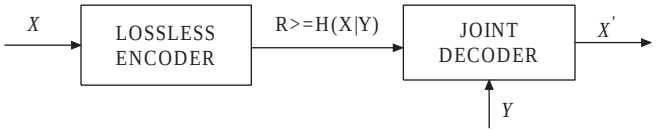


Figure 4.3. Lossless source coding with side information at the decoder.

4.1.2 Wyner-Ziv Coding

The Slepian-Wolf scheme focused on lossless source coding of discrete sources with side information at the decoder. However most sensor network applications deal with continuous sources. The rate distortion with side information at the decoder thus becomes a big concern. The problem to solve in the lossy source coding is how many bits are needed to encode X under the constraint that the average distortion between X and X' is $E[d(X, X')] \leq D$, assuming the side information Y is available only at the decoder.

Wyner and Ziv [55] first considered this problem and gave the rate-distortion function $R^*_{WZ}(D)$ for both discrete and continuous cases and general distortion metrics $d(.)$. Fig. 4.4 is an illustration of Wyner-Ziv coding. In general, Wyner-Ziv coding set up the Slepian-Wolf coding in that coding of X is with respect to a fidelity criterion rather than lossless coding.

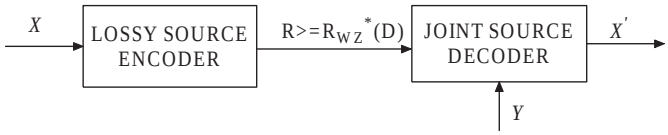


Figure 4.4. Wyner-Ziv coding or lossy source coding with side information.

But the important thing about Wyner-Ziv coding is that it normally suffers rate loss when compared to lossy coding of X as the side information Y is available at both the encoder and decoder. One exception is when X and Y are jointly Gaussian which is of special interest in practice since many image and video sources can be modeled as jointly Gaussian.

Since distortion is introduced to the source with Wyner-Ziv coding, quantization is needed in source coding. Usually there is still certain correlation in the quantized version of X and the side information Y, thus Slepian-Wolf coding could be employed to reduce the rate. In this case, the side information Y is used in jointly decoding and estimating X' at the decoder to help reduce the distortion $d(X, X')$ for non-binary sources. Fig 4.5 depicts the block diagram of a generic Wyner-Ziv coder.

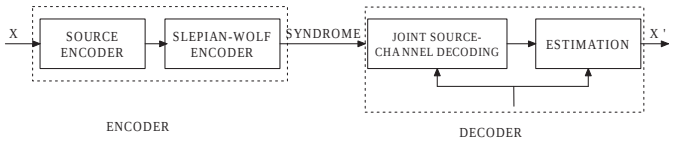


Figure 4.5. Block diagram of a generic Wyner-Ziv coder.

4.2 Intuition behind approch

The above section discussed lossless (Slepian-Wolf) and lossy (Wyner-Ziv) source coding with side information available only at the decoder. Most of the work in DSC so far has been focusing on the two problems. In wireless sensor network, employing current

DSC schemes requires the sensor nodes transmitting correlated information to cooperate in a small group so that one node provides side information and the others compress the information down to the Slepian-Wolf or the Wyner-Ziv limit.

Tha major concern for practical application of DSC is the correlation model. Theoretically, two correlated non-binary sources can be constructed easily. An example with uniform distribution is shown as follows:

- Let $X = X_0X_1...$ and $Y = Y_0Y_1...$ be two correlated non-binary sequences taking values in $[L, R]$.
- Generate the $i.i.d$ sequence X using the probability distribution $P(X_k = i) = 1/(R - L)$ where $i \in [L, R]$.
- Define the sequence Y from the sequence X using the conditional probability distribution $P(Y_k = j|X_k = i) = p_{ij}$, where $i, j \in [L, R]$. The joint probability distribution between sources will be denoted by $P(X_k = i, Y_i = j) = p_{ij}/(R - L)$.

Although significant effort has been put in DSC design for various correlation models, in real sensor networks there still exist many situations that are hard to determine joint probability functions. For instance, the correlation statistics of the video surveillance networks can be mainly a function of the sensors' location. Fig 4.6 is another example of the noisy versions of the acoustic signal strength collected using the Xbow wireless sensor network professional developer's kit MOTE-Kit.

This paper addresses the issue of lossy coding for correlated non-binary sources in the Xbow wireless sensor networks. We are interested in the measurement noise in a wireless sensor network, specifically in the Chief Executive Officer (CEO) problem [57]. In this particular application, the CEO of a company employs a number of agents to observe an event and each of the agents provides the CEO with his/her noise version of the event. The agents are not allowed to convene, and the goal of the CEO is to recover as much information as possible about the actual event from the noisy observations received

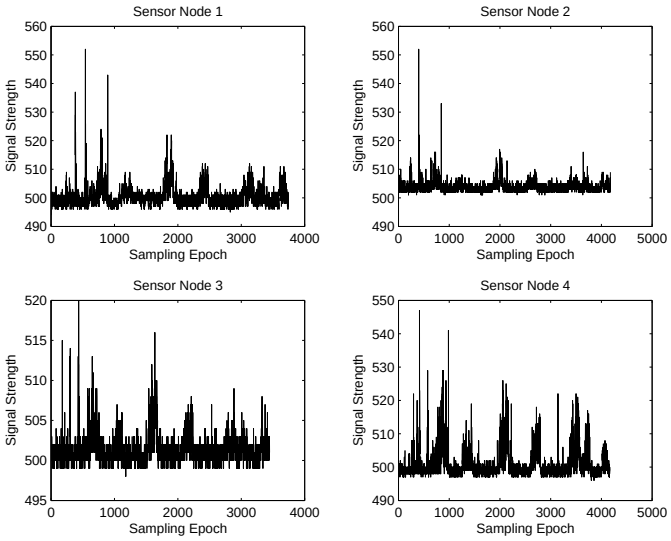


Figure 4.6. Four Noisy observations of acoustic signal strength.

from the agents, while minimizing the total information rate from the agents. The CEO problem can then illuminate the measurement noise at the sensor node.

Preliminary practical code constructions for the CEO problem appeared in [58,59], based on the Wyner-Ziv coding approaches, but they are only limited to special cases. Fig 4.7 is a CEO example in a wireless sensor network where the central hub node is responsible to recover the information from the noisy measurements.

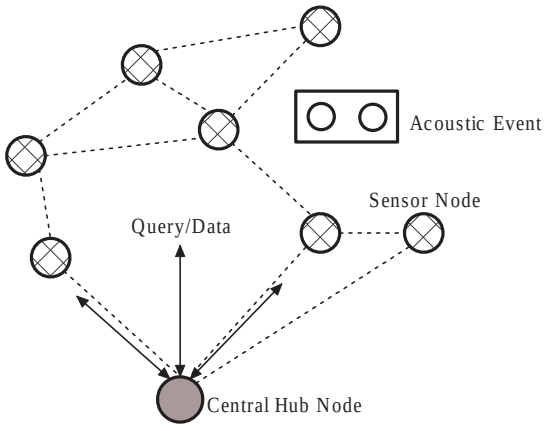


Figure 4.7. A CEO example of sensor network.

4.3 Construct Codeword Cosets

For correlated binary sources X and Y, Y is a noise corrupted version of X as $Y = X + N$, where N is an additive Gaussian noise. The correlation between the interested output X and the side information Y can be modeled with a "virtual" correlation channel, then a good channel code over this channel can provide us with a good Slepian-Wolf codes. In a sense, the seemingly source coding problem of Slepian-Wolf coding can be considered as a channel coding problem.

This section details our spectrum efficient coding scheme for correlated non-binary sources in wireless sensor networks. For interested information X, the encoder side consists of two parts: source encoder and coset encoder. We apply Lloyd-Max quantization in souce encoder which conducts the design of the initial codebook. The non-binary sources are then represented by the binary codewords according to the quantization levels. A coset encoder is constructed to save transmitting bits over channels. A n-bit codeword is transmitted by a m-bit ($m < n$) coset leader to achieve a compression ratio of $n : m$ after the coset encoder. Side information Y will be transmitted at full rate, i.e. not through the coset encoder. The block diagram of our coding scheme is illustrated in Fig 4.8.

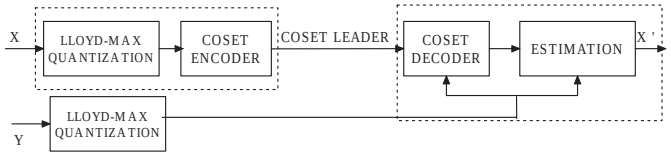


Figure 4.8. Block diagram of the asymmetric coding scheme.

We next give an example of constructing the coset encoder.

Example 4: Construct Codeword Cosets with Hamming Distance $d_H = 3$

For 8-level Lloyd-Max quantization, the input to the coset encoder is a 3-bit binary codeword $X_Q \in [000, 001, 011, 010, 110, 111, 101, 100]$. Assuming the Hamming distance between X_Q and the quantized binary side information Y_Q is $d_H(X_Q, Y_Q) \leq 1$, the cosets for X_Q can be constructed using the parity-check matrix H

$$H = \begin{bmatrix} 1\ 1\ 0 \\ 1\ 0\ 1 \end{bmatrix} \tag{4.1}$$

Four coset sets are constructed as $C1 = [000, 111]$, $C2 = [001, 110]$, $C3 = [010, 101]$ and $C4 = [011, 100]$. The transmitted coset leader X_C is associated with the syndrome $s = X_Q H^T$. Sending the 2-bit coset leader instead of the original 3-bit X_Q achieves a compression ratio of $3 : 2$.

Now consider the noisy observation from sensor node 1 (see Fig 4.6 (a)) as the interested information X. Results from 8-level Lloyd-Max quantization are presented in Table 4.1.

Table 4.1. Results from 8-level Lloyd-Max Quantization

Codebook	Occurring Probability	Binary Codebook
498.09	0.4923	000
500.37	0.2809	001
503.06	0.1590	011
507.3	0.0457	010
511.31	0.0136	110
515.26	0.0051	111
523	0.0027	101
544	0.0008	100

From Table 4.1, the first codeword after quantization 498.09 occurs at a dominant probability of 49.23%. The probability of occurence decreases dramatically along the

initial codebook. The binary codewords are assigned such that along the probability decreasing, every adjacent codeword differs in only 1 bit.

Suppose sensor node 3 (see Fig 4.6 (c)) is transmitting the side information Y for decoding. Data from sensor node 3 is quantized separately using a Lloyd-Max quantizer. Now we have X_Q and Y_Q at 3-bit correlated binary codewords. Perfect coset encoder [8] requires that X_Q and Y_Q are correlated in the way that the Hamming distance between X_Q and Y_Q is no more than one. Then the cosets for X_Q are constructed that the elements within each coset have maximal Hamming distance $d_H = 3$ as depicted in example 4.

In this work, the correlation between X_Q and Y_Q is unknown or can hardly reach the perfect correlation. But with the knowledge of the codewords probability distribution, the coset construction could be done in a different way.

We propose to design the coset sets minimizing the overall *cross ratio*. The *cross ratio* is defined as the ratio that within one coset, the codeword with less occuring probability will cross the other. We intend to decrease the decoding failure by reducing the *cross ratio* while keeping the Hamming distance within each coset as large as possible. Table 4.2 gives the *cross ratio* of two different coset sets.

Table 4.2. Collision Ratio of Two Cosets Sets

Coset Set 1	Cross Ratio	Coset Set 2	Cross Ratio
$(000, 111)$	0.01	$(000, 110)$	0.027
$(001, 110)$	0.046	$(001, 111)$	0.018
$(011, 100)$	0.01	$(011, 101)$	0.017
$(010, 101)$	0.056	$(010, 100)$	0.017
Overall	0.0305	Overall	0.01975

From Table 4.2, we see that coset set 2 has less *cross ratio* even though the Hamming distance within each coset is $d_H = 2$ but not 3.

The parity-check matrix to construct the codeword coset 2 with Hamming distance $d_H = 2$ is shown as bellow:

$$H = \begin{bmatrix} 0\ 0\ 1 \\ 1\ 1\ 0 \end{bmatrix} \tag{4.2}$$

At the decoder, we use the side information Y_Q to look for the most-likely codeword from the coset represented by the transmitted coset leader. The decoder then get the optimal estimation of X using all received information.

4.4 Simulation Results

Simulations are performed over the acoustic noisy observations from the Xbow wireless sensor network professional developer's kit MOTE-Kit. 8 sets of acoustic noisy version are collected from 8 distributed deployed sensors in a lab. Information from the sensor closest to the acoustic source is set as side information for decoding. All others are encoded separately and reconstructed at the decoder with the side information. The correlation-in-dB between the interested information and the side information is presented in Table 4.3.

Table 4.3. Correlation-in-dB between X and Y

Sensor Node	Correlation-in-dB
1	5.7988
2	5.0864
3	6.3903
4	4.2262
5	4.2343
6	4.1522
7	5.5238

Due to the packet loss in data collecting at the central hub node, the correlation between the interested information X and the side information Y from sensor node 8 is pretty low. We choose two sensor nodes (node 1 and node3) with the highest correlation to side information Y for our simulation.

For comparison, we generate two ideal $i.i.d$ Gaussian sequences X and Y correlated by $Y = X+N$, where X has zero mean and unit variance and N is the zero mean Guassian noise with variance σ_N. Y, the corrupted version of X is the side information used for joint decoding.

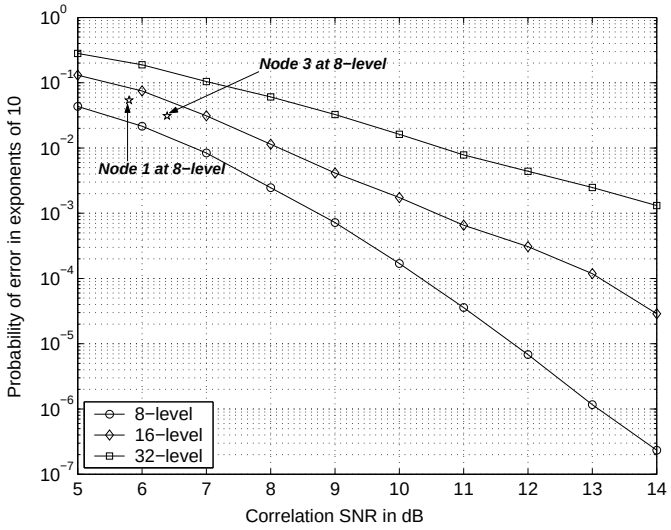


Figure 4.9. Probability of Error for R=2bits/sample.

We employ 8-level, 16-level and 32-level Lloyd-Max quantization. Each is partitioned into two cosets, where each coset set contains 2, 4 and 8 codewords respectively. The number of samples used for the Monte Carlo simulations is 10^7. Fig. 4.9 shows the probability of decoding error for the above three schemes and normalized distortion with correct decoding only is plotted versus correlation SNR for the same schemes in Fig. 4.10. Observe that for a given correlation SNR, as the number of quantization levels increases,

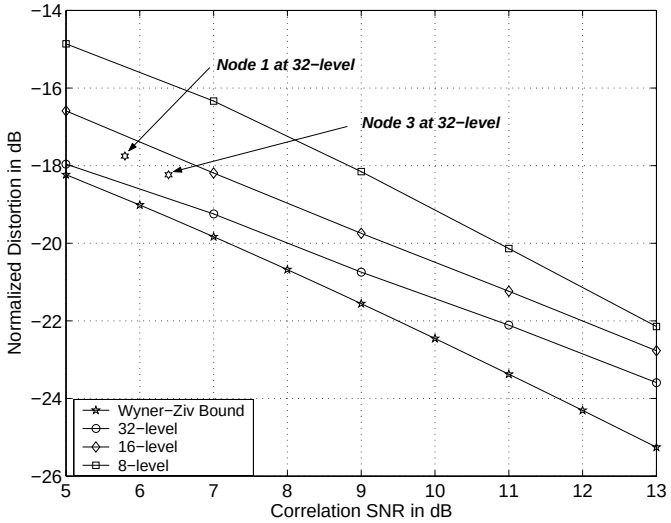


Figure 4.10. Normalized Distortion for R=2bits/sample.

the normalized distortion decreases and the probability of decoding error increases. Ideally for a given transmission rate, we want to quantize with a large number of levels to cut down distortion, but the tradeoff between the distortion and probability of decoding errors put a constraint in this. As can be noted from Fig. 4.9 and 4.10, at 8-level quantization, performance of sensor readings from node 1 and node 3 are approximately 0.5 dB from the one of ideal $i.i.d$ Gaussian sources.

In coset encoding, we compare different coset construction methods. Fig. 4.11 gives the result of coset set 1 and coset set 2 at 8-level quantization. The performance of coset set 2 is slightly better than the one of coset set 1.

Last the coding scheme is employed to all 7 sensor nodes and compute the actual transmitted data bits. We process the information observed in the common epoch and discard the incomplete observations. Under the scheme of two coset sets, the real transmitted data bits remain the same for 8-,16- and 32-level cases. Results are presented in Table 4.4.

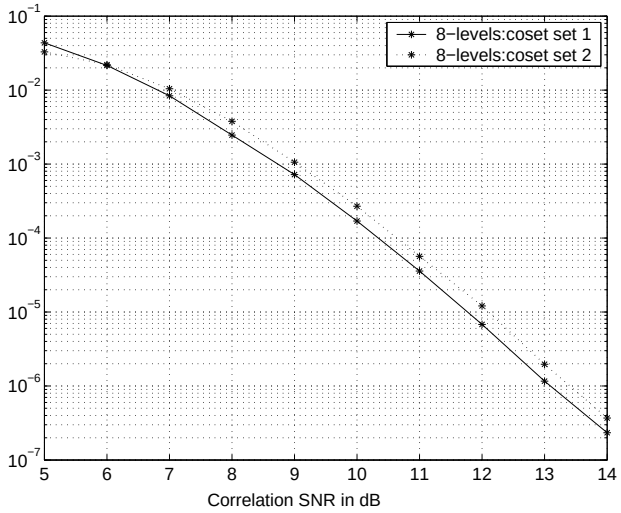


Figure 4.11. Probability of Error for R=2bits/sample.

Table 4.4. Compression Ratio of real transmitted data

Levels	Original bits	Transmitted bits	Compression Ratio
8-level	78339	31815	2.46
16-level	104532	31815	3.29
32-level	130665	31815	4.11

CHAPTER 5

DATA FUSION IN A MULTI-TARGET RADAR SENSOR NETWORK

5.1 Introduction

Wireless sensor networks (WSN) have attracted growing interest in various applications, especially in the area of battlefield surveillance, health care and telemedicine, environmental and habitat monitoring. Radar as a powerful sensor system, has been employed for the detection and location of reflecting objects such as aircraft, ships, vehicles, people and natural environment. By radiating energy into space and detecting the echo signal reflected from an object or target, the radar system can determine the presence of a target. Furthermore, by comparing the received echo signal with the transmitted signal, the location of a target can be determined along with other target related information [23].

Conventional radar system operates as a pure independent entity. While in a resource-constrained WSN, such detached operation may lead to deteriorated performance and waste of limited resources. Collaborative signal and information processing over the network is a very promising area of research and is related to distributed information fusion [24]. Important technical issues include the degree of information sharing between sensors and how sensors fuse the information from other sensors. Processing data from more sensors generally results in better performance but also requires more communication resources. Similarly, less information is lost when communicating information at a low level (e.g., raw data), but requires more bandwidth. Therefore, it is a tradeoff between system performance and resource utilization in collaborative information processing and data fusion.

A lot of prior research in data fusion is based on the assumption of lossless communication, i.e., the information sent from local sensors is perfectly recovered at the fusion center. For example, in [25] and [26], Vashney *et. al* investigated the optimum fusion rules under the conditional independence assumption. Other papers [27, 28] addressed the problem of distributed detection with constrained system resources, most of which provided the solutions to optimize sensor selection. However, this lossless communication assumption is not practical for many WSNs where the transmitted data suffers from channel fading and multi-user interference. On the other hand, decision fusion with non-ideal communication channels are studied at both fusion center level [29, 30] and at the sensor level [31, 32]. In [30], Thomopoulos and Zhang derived the optimal thresholds by assuming a simple binary symmetric channel between sensors and the fusion center. Their method is quite simple but requires global knowledge of the entire system. In [29], channel-aware decision fusion rules have been developed using a canonical distributed detection system where binary decisions from multiple parallel sensors are transmitted through fading channels to a fusion center. Later, Lin *et. al* [33] have extended the channel aware decision fusion rules to more realistic WSN models that involve multi-hop transmissions. The above results, however, are mostly obtained based on one target or one event detection which is not applicable to multi-target situations. Furthermore, in a radar sensor system, when clutter, the unwanted echoes from the natural environment is much larger than the receiver noise, detection can be quite difficult from that when the noise is dominant.

The objective of this work is to derive the decision fusion rules of multiple fluctuating targets in multi-radar (MT-MR) sensor networks. We focus on the detection decision performance of fused data with the existence of clutter. The theoretical formulation of the MIMO decision fusion problems is presented. It is assumed that the multiple targets are stationary targets embedded in clutter. Rayleigh target fluctuation

model and Gaussian clutter are used in the first stage study. Particularly, we assume that the radar in our scenario, is a constant false alarm receiver (CFAR) when receiving. CFAR automatically raises the threshold level to keep clutter echoes and external noise from overloading, which performs as a good rejection of clutter.

The remainder of this paper is organized as follows. Next section introduces the concept of clutter and target fluctuation model in radar sensor system. Section 5.3 briefly overview the previous work on fusion rules designed for a canonical parallel distributed detection system with single hop transmission between sensor nodes and fusion center. Section 5.4 presents our MIMO decision fusion model for multi-target multi-radar sensor networks. Simulation and performance analysis are presented in Section 5.5. Section 5.6 concludes this paper.

5.2 Target Detection in Radar Sensor System

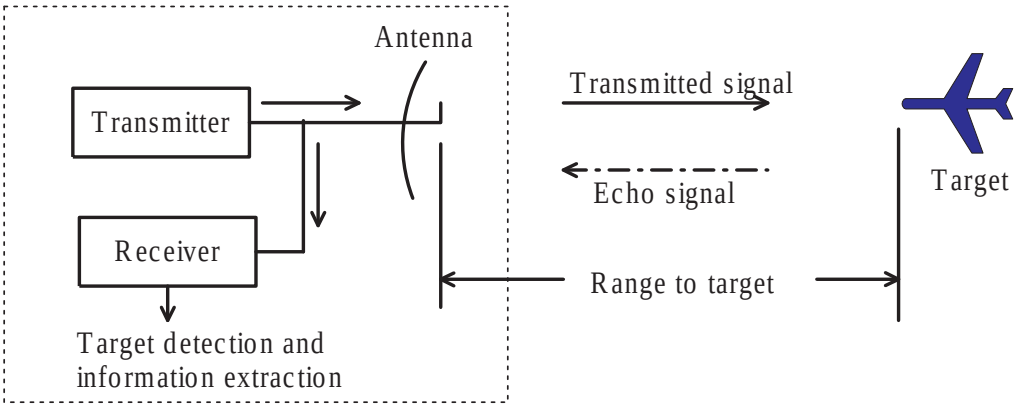


Figure 5.1. Basic Principle of Radar System.

The basic principle of radar [23] is illustrated in Fig. 5.1. An electromagnetic signal is generated by the transmitter and is radiated into space by the antenna. A portion of the transmitted energy is intercepted by the target and reradiated in various directions.

The reradiation directed back towards the radar is collected by the radar antenna, which delivers it to a receiver. There it is processed to detect the presence of the target and determine its location. In active radar sensor networks, the received data usually consists of three parts: white thermal noise, clutter scattered by the land environment, and if a target is present, a reflected or reradiated version of the transmitted signal [60]. That is, $y^{(t)} = \alpha^{(t)} s^{(t)} + n^{(t)} + \omega^{(t)}$, in which $s^{(t)}$ and $y^{(t)}$ are the transmitted and received signals, respectively. The values $\alpha^{(t)}$ is the target cross section or radar cross section (RCS). It is assumed that $n^{(t)}$ is additive noise and $\omega^{(t)}$ is the returned clutter, a distorted version of the transmitted signal $s^{(t)}$. In the work presented here, it is assumed that the received clutter is much larger than the white thermal noise, i.e. $\omega^{(t)} >> n^{(t)}$. Thus it becomes $y^{(t)} \approx \alpha^{(t)} s^{(t)} + \omega^{(t)}$, when $\omega^{(t)} >> n^{(t)}$.

The classical radar equation uses the target cross section or radar cross section (RCS) to determine the power density returned to the radar for a particular power density incident on the target. Nevertheless, the scattering of electromagnetic energy from a target is a rather complicated phenomenon, which depends on a number of factors such as target geometry, size, shape, aspect, altitude with respect to the radar antenna etc. Therefore, it has been advantageous to model the target RCSs as a random variable. Some common fluctuation models are now available in the open literature, i.e. Swerling chi , lognormal, Rayleigh, Weibull as a compound Rayleigh distribution, Shadowed Rice target etc. This work treats the target fluctuation as Rayleigh distribution which has the probability density function (pdf) as $f_\nu(\nu) = \frac{\nu}{\sigma_c^2} \exp\left(-\frac{\nu^2}{2\sigma_c^2}\right)$, where $2\sigma_c^2$ is the mean square value of the envelope ν.

Clutter is the unwanted echoes from the natural environment such as land, sea, rain, birds, insects etc. Clutter can be spatially distributed so that it is much larger in physical size than the radar resolution cell. There are also point or discrete clutter echoes that produce large backscatter. Because of the highly variable nature of clutter

echoes it is often described by a probability density function. Some forms of clutter have similar distributions as the target fluctuation model, e.g., Gaussian, Rayleigh, log-Normal and Weibull. Nevertheless, other distributions have been proposed to describe the special statistics of clutter including K-distribution, contaminated normal, gamma and log-Weibull. For the first stage of this work, we set the returned clutter to follow the Gaussian distribution with zero mean.

5.3 Review of Previous Decision Fusion Rules

In a single target, single hop sensor network, the typical parallel fusion structure in a flat fading channel is depicted in Fig. 5.2. The received signal at the fusion center from kth sensor is $y_k = h_k u_k + n_k$, where h_k is the channel fading envelope and n_k is the zero-mean additive Gaussian noise with variance σ^2. K sensors collect data generated according to either H_0 (there is no target present) or H_1 (there is target present) and transmit these decisions over fading and noisy channels to a fusion center. The fusion center tries to decide which hypothesis is true based on the received data y_k from all k.

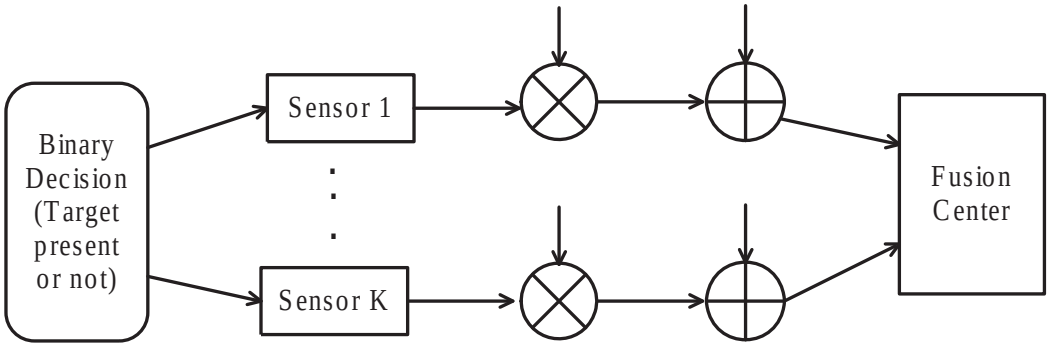


Figure 5.2. Single-target, single-hop decision fusion model.

Assume that the kth local sensor makes a binary decision $u_k \in \{+1, -1\}$, with false alarm and detection probability P_{fk} and P_{dk} respectively. That is, we have $P_{fk} =$

$P[u_k = 1|H_0]$ and $P_{dk} = P[u_k = 1|H_1]$. Several decision fusion rules have been developed based on the above model in [33]. Throughout this work, we use $\Lambda^{(s)}$ to denote the fusion statistics for the single hop, single target transmission model.

- Optimal LR-based fusion statistic using complete prior knowledge. Assuming complete channel knowledge, the optimal LR-based fusion statistic was derived as

$$\Lambda_1^{(s)} = \prod_{k=1}^{K} \frac{P_{dk}\psi_k^{(+)} + (1 - P_{dk})\psi_k^{(-)}}{P_{fk}\psi_k^{(+)} + (1 - P_{fk})\psi_k^{(-)}} \tag{5.1}$$

  where $\psi_k^{(+)} = e^{-((y_k - h_k)^2/2\sigma^2)}$ and $\psi_k^{(-)} = e^{-((y_k + h_k)^2/2\sigma^2)}$ and $Y = [y_1, ..., y_k]^T$ is a vector containing observations received from all K sensors.

- LR-based fusion rules using only fading statistics for Rayleigh fading channel. Implementing the optimal LR test as in (5.1) requires that all a priori information, including the instantaneous channel gains. Under the Rayleigh fading model, the LR-based fusion statistic using only the fading parameter is summarized below

$$\Lambda_2^{(s)} = \prod_{k=1}^{K} \frac{P_{dk}\Psi_k^{(+)} + (1 - P_{dk})\Psi_k^{(-)}}{P_{fk}\Psi_k^{(+)} + (1 - P_{fk})\Psi_k^{(-)}} \tag{5.2}$$

  where $\Psi_k^{(+)} = 1 + \sqrt{2\pi\gamma}y_k e^{(\gamma^2 y_k^2/2)}Q(-y_k\gamma)$, $\Psi_k^{(-)} = 1 - \sqrt{2\pi\gamma}y_k e^{(\gamma^2 y_k^2/2)}Q(y_k\gamma)$ and $\gamma = (\sigma_c/\sigma_n\sqrt{\sigma_c^2 + \sigma_n^2})$ with $2\sigma_c^2$ being the mean square value of the fading channel, σ_n^2 is the noise variance, and $Q(\cdot)$ is the complementary distribution function of a standard Gaussian random variable.

- A two-stage approximation using the Chair-Varshney fusion rule. A direct alternative to the above LR-based fusion rules is to consider the information transmission and decision fusion as a two-stage process: first y_k is used to infer about u_k: then, the estimation of u_k are employed in the optimum fusion rule. Given the model in Fig. 5.2, the maximum likelihood (ML) estimation for u_k is $\hat{u_k} = sign(y_k)$. Ap-

plying the fusion rule derived in [33], the Chair-Varshney fusion rule is obtained as

$$\Lambda_3^{(s)} = \sum_{y_k<0} \log\left(\frac{1-P_{dk}}{1-P_{fk}}\right) + \sum_{y_k>0} \log\left(\frac{P_{dk}}{P_{fk}}\right) \tag{5.3}$$

- Fusion statistics using a maximum ratio combiner (MRC). In the low SNR regime, if the local sensors are identical, i.e., P_{dk} and P_{fk} are the same for all ks, then $\Lambda_1^{(s)}$ reduces to a form analogous to an MRC

$$\Lambda_4^{(s)} = \frac{1}{K}\sum_{k=1}^{K} h_k y_k \tag{5.4}$$

- Fusion statistics using an equal gain combiner (EGC). In the low SNR regime, if the local sensors are identical, i.e., P_{dk} and P_{fk} are the same for all ks, then $\Lambda_2^{(s)}$ reduces to a form analogous to an EGC

$$\Lambda_5^{(s)} = \frac{1}{K}\sum_{k=1}^{K} y_k \tag{5.5}$$

Among the above five fusion rules, $\Lambda_1^{(s)}$ requires complete channel knowledge and provides uniformly the most powerful detection performance. At low SNR, the MRC statistic provides the best performance among the three suboptimum fusion rules; while at high SNR, the Chair-Varshney fusion rule outperforms the MRC and the EGC statistics. The EGC statistic, however, provides better performance over a wide range of SNR than the MRC statistic and the Chair-Varshney fusion rule and requires the least amount of prior information.

5.4 MIMO decision fusion model for multi-target sensor networks

In the scenario, it is assumed that there are multiple radar sensors and multiple stationary targets in the field. A radar detects the presence of a target and generates the decision data according to two hypothesis: H_0 : there is no target present and H_1 : there is target present. Each decision data is transmitted to the fusion center, normally a radar sensor as well. In a multi-hop radar sensor network, the decision data is relayed via several radars to reach the fusion center. When there are multiple radar sensors and multiple targets in the field, the data fusion problem can be roughly modeled as a Multi-Input Multi-Output (MIMO) fusion problem. This paper assumes the radar sensors are disparate, geographically dispersed in the field such that the radar observations or decisions are spatially independent. Fig. 5.3 illustrates an example of single-hop decision fusion problem.

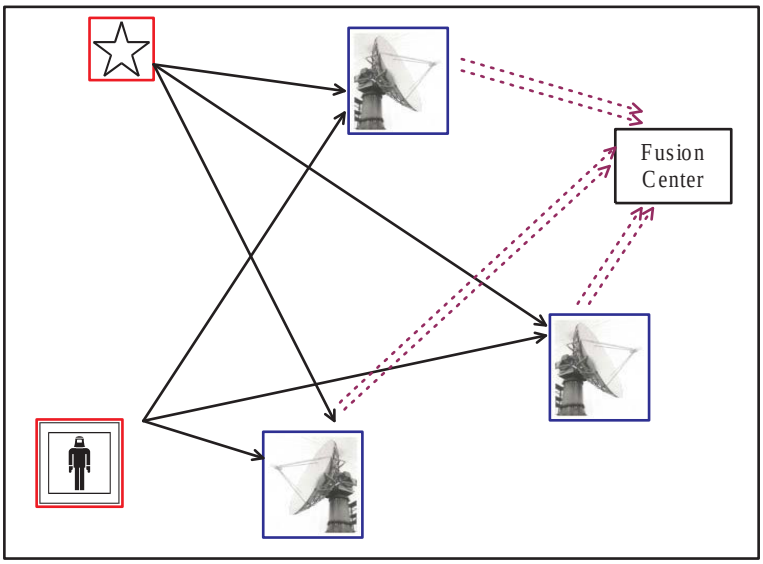


Figure 5.3. MIMO fusion model.

Let M denote the number of radar sensors and N be the number of targets. The received signal $Y^{(t)}$ at the fusion center at time t is a $N \times M$ matrix.

$$\begin{bmatrix} y_{11}^{(t)} & \cdot & \cdot & \cdot & y_{M1}^{(t)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ y_{1N}^{(t)} & \cdot & \cdot & \cdot & y_{MN}^{(t)} \end{bmatrix} \tag{5.6}$$

It is assumed that the radar sensors are geographically dispersed and detection decisions are made at each separate local radar. The element $y_{ij}^{(t)}$ of (5.6) is the decision (target present or absent) of the jth target from the ith radar sensor. The value, $y_{ij}^{(t)}$, can be represented as

$$y_{ij}^{(t)} = \alpha_{ij}^{(t)} \cdot {s_{ij}}^{(t)} + \omega_{ij}^{(t)} \tag{5.7}$$

Observe that in [33], the researchers assume that both the false alarm P_f and probability of detection P_d are fixed and identical for all local sensors. Moreover there is no correlation between the false alarm P_f and probability of detection P_d. In radar system however, this assumption is very impractical especially in the heavy clutter situation. One method to suppress the heavy clutter is to use a *constant false alarm rate* (CFAR) receiver. The CFAR automatically raises the threshold level to keep clutter echoes and external noise from overloading the automatic tracker with extraneous information. In the study, we assume the receivers of all radar sensors are CFAR which implies that though the false alarm rate is a constant, the probability of detection of each local radar sensor varies. We use P_f as the fixed false alarm rate and P_{di} to denote the distinct probability of detection at radar sensor i throughout this work.

We next derive the MIMO decision fusion rules for the multi-target radar sensor networks starting from the single-hop radar sensor networks.

5.4.1 Decision fusion rule in single-hop radar sensor networks

- Assume we have complete knowledge of the target fluctuation coefficients, the optimal LR-based fusion rule to detect a single target for example, the jth target was derived as

$$\Lambda_j^{(1)} = \prod_{i=1}^{M} \frac{P_{di}\psi_{ij}^{(+)} + (1-P_{di})\psi_{ij}^{(-)}}{P_f\psi_{ij}^{(+)} + (1-P_f)\psi_{ij}^{(-)}} \quad j = 1, ..., N \tag{5.8}$$

where

$$\underline{\psi}_i^{(\pm)} = \begin{bmatrix} e^{-(y_{i1} \mp \alpha_{i1})^2/2\sigma^2} \\ e^{-(y_{i2} \mp \alpha_{i2})^2/2\sigma^2} \\ \cdot \\ \cdot \\ \cdot \\ e^{-(y_{iN} \mp \alpha_{iN})^2/2\sigma^2} \end{bmatrix} \qquad i = 1, ..., M \tag{5.9}$$

From Eq.(5.1), the decision on a single target is the fusion result based on the observations from all M radar sensors. A complete decision vector for N targets are denoted as $\underline{\Lambda}^{(1)} = [\Lambda_1, \Lambda_1, ..., \Lambda_N]^T$

- LR-based fusion rules using only target fluctuation statistics. Under the assumption of Gaussian clutter model and Rayleigh target fluctuation model, the LR-based fusion statistic using only the target fluctuation coefficients is summarized below

$$\Lambda_j^{(2)} = \prod_{i=1}^{M} \frac{P_{di}\Psi_{ij}^{(+)} + (1-P_{di})\Psi_{ij}^{(-)}}{P_f\Psi_{ij}^{(+)} + (1-P_f)\Psi_{ij}^{(-)}} \quad j = 1, ..., N \tag{5.10}$$

where $\gamma = (\sigma/\sigma_\omega\sqrt{\sigma^2+\sigma_\omega^2})$ with $2\sigma^2$ being the mean square value of the target fluctuation model, σ_ω^2 is the clutter variance.

$$\underline{\Psi}_i^{(\pm)} = \begin{bmatrix} 1 \pm \sqrt{2\pi\gamma} y_{i1} e^{(\gamma^2 y_{i1}^2/2)} Q(\mp y_{i1}\gamma) \\ 1 \pm \sqrt{2\pi\gamma} y_{i2} e^{(\gamma^2 y_{i2}^2/2)} Q(\mp y_{i2}\gamma) \\ . \\ . \\ . \\ 1 \pm \sqrt{2\pi\gamma} y_{iN} e^{(\gamma^2 y_{iN}^2/2)} Q(\mp y_{iN}\gamma) \end{bmatrix} \qquad i = 1, ..., M \tag{5.11}$$

- Fusion statistics using a maximum ratio combiner (MRC). In the low SNR regime, if the local radar sensors are identical,i.e., P_{di} and P_{fi} are the same for all is, then $\underline{\Lambda}^{(1)}$ reduces to a form analogous to an MRC

$$\Lambda_j^{(3)} = \frac{1}{M}\sum_{i=1}^{M} \alpha_{ij} y_{ij} \qquad j = 1, ..., N \tag{5.12}$$

- Fusion statistics using an equal gain combiner (EGC). At low SNR regime, if the local radar sensors are identical, i.e., P_{di} and P_{fi} are the same for all ks, then $\underline{\Lambda}^{(2)}$ reduces to a form analogous to an EGC

$$\Lambda_j^{(4)} = \frac{1}{M}\sum_{i=1}^{M} y_{ij} \qquad j = 1, ..., N \tag{5.13}$$

5.4.2 Decision fusion rule in multi-hop sensor networks

When considering a multi-hop radar sensor networks, decision fusion problem for multiple targets could be quite complicate. For simplicity, we follow the assumption in single-hop case and assume the relay radar sensors have no direct observation of all the targets. Comparing Fig. 5.3 with Fig. 5.4, the above assumption assures that the MIMO

fusion problem for multi-hop remains the same dimension as the one in the single-hop case.

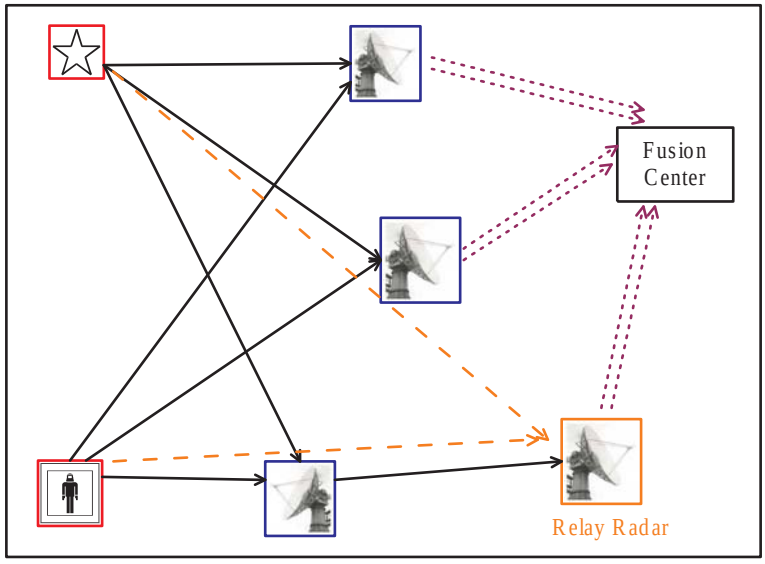


Figure 5.4. MIMO fusion model for multi-hop radar sensor networks.

We make the further assumption that each relay radar makes a simple hard decision on the signal transmitted from its last hop radar. Therefore, given that the clutter is Gaussian, we have

$$s^k = sign(\alpha^{k-1} s^{k-1} + \omega^{k-1}) \tag{5.14}$$

Hence, the ultimate received signals at the fusion center transmitted from all the M last hop radars have the similar form as (5.6). We also assume that the Rayleigh RCS has unit power, i.e., $E[\alpha_{ij}^2] = 1$ and Gaussian clutter has variance σ^2 to facilitate SCR calculation later in the paper.

Implementing the decision rules for single target, multi-hop WSNs in [33] to our multi-target, multi-hop radar sensor networks, we get the decision fusion rules as follows.

- Optimal LR-based Fusion Rule

  In multi-hop radar sensor network, we assume only the first hop radar sensors are CFAR with false alarm P_f^0 and probability of detection P_{di}^0. Let P_{di}^c be the probability of detection at the ith radar in the last relay, [33] has proved that for one given target detection, $P_{di}^c \approx P_{di}^0$ and $P_f^c = P_f^0$ at high signal to clutter ratio (SCR). At low SCR, P_{di}^c and P_f^ccan be approximated as

$$P_{di}^c \approx \frac{1}{2} + \frac{2^{M_i}(\prod_{k=0}^{M_i-1}\alpha_{ik})}{(\sqrt{2\pi}\sigma)^{M_i}}\left(P_{di}^0 - \frac{1}{2}\right) \tag{5.15}$$

$$P_f^c \approx \frac{1}{2} + \frac{2^{M_i}(\prod_{k=0}^{M_i-1}\alpha_{ik})}{(\sqrt{2\pi}\sigma)^{M_i}}\left(P_f^0 - \frac{1}{2}\right) \tag{5.16}$$

  Assume there are M radar sensors in the last hop, M_i is the number of hops at the ith radar. α_{ik} is RCS value at the kth relay of radar i.

  The optimum LR-based fusion rule for multi-target radar sensor networks can be written as

$$\Lambda_j^{(1)} = \prod_{i=1}^{M} \frac{P_{di}^c + (1 - P_{di}^c)\cdot\phi_{ij}}{P_f^c + (1 - P_f^c)\cdot\phi_{ij}} \qquad j = 1, ..., N \tag{5.17}$$

  where

$$\underline{\phi}_i = \begin{bmatrix} e^{-(2y_{i1}\alpha_{i1})^{M_i}/\sigma^2} \\ e^{-(2y_{i2}\alpha_{i2})^{M_i}/\sigma^2} \\ \cdot \\ \cdot \\ \cdot \\ e^{-(2y_{iN}\alpha_{iN})^{M_i}/\sigma^2} \end{bmatrix} \qquad i = 1, ..., M \tag{5.18}$$

- Denote $\underline{\Lambda}^{(2)}$ as the LR rule that corresponds to the case when only the target RCS statistics are known. $\underline{\Lambda}^{(2)}$ can be derived for the multi-hop MT-MR sensor networks.

$$\Lambda_j^{(2)} = \prod_{i=1}^{M} \frac{1 + [P_{di}^c - Q(\gamma y_{ij})]\sqrt{2\pi}\gamma y_{ij} e^{(\gamma y_{ij})^2/2}}{1 + [P_f^c - Q(\gamma y_{ij})]\sqrt{2\pi}\gamma y_{ij} e^{(\gamma y_{ij})^2/2}} \quad j = 1, ..., N \tag{5.19}$$

  where $\underline{y}_i = [y_{i1}, y_{i2}, ...y_{iN}]$ is a vector containing all N decision data from radar i. P_{di}^c and P_f^c are denoted as in (5.15) and (5.16).

- Decision fusion rules of Maximum ratio combiner (MRC) and equal gain combiner (EGC) have the identical format as the single hop case because for both of them, the decision fusion only depends on the last hop.

5.5 Simulation Results

This section simulates the performance of the decision fusion rules derived for multi-target radar sensor networks. For ease of SCR calculation, we assume that all the target RCSs have unit power, i.e, $E[\alpha_{ij}^2] = 1$. Binary decisions are made at the local radar sensors and the relay radars. The target RCS are generated using the Rayleigh model.

For multi-target, single-hop radar sensor network and multi-target, multi-hop radar sensor network, we are interested to compare the four decision fusion rules:

- Optimal LR-based rule
- LR-based rule with target RCS statistics only
- MRC rule
- EGC rule

In all simulations, we assume the constant false alarm rate $P_f = 0.01$ (for multi-hop case, $P_f = 0.01$ is the one at the first hop). Under hypothesis H_0 when a target is detected as absent, $P_f = Q(\frac{X_t}{\sigma})$. We then know the detection threshold $X_t = Q(P_f)^{-1}\sigma$. When a target is detected, i.e., hypothesis H_1, the probability of detection $P_{di} = Q(\frac{X_t - \alpha_i}{\sigma})$.

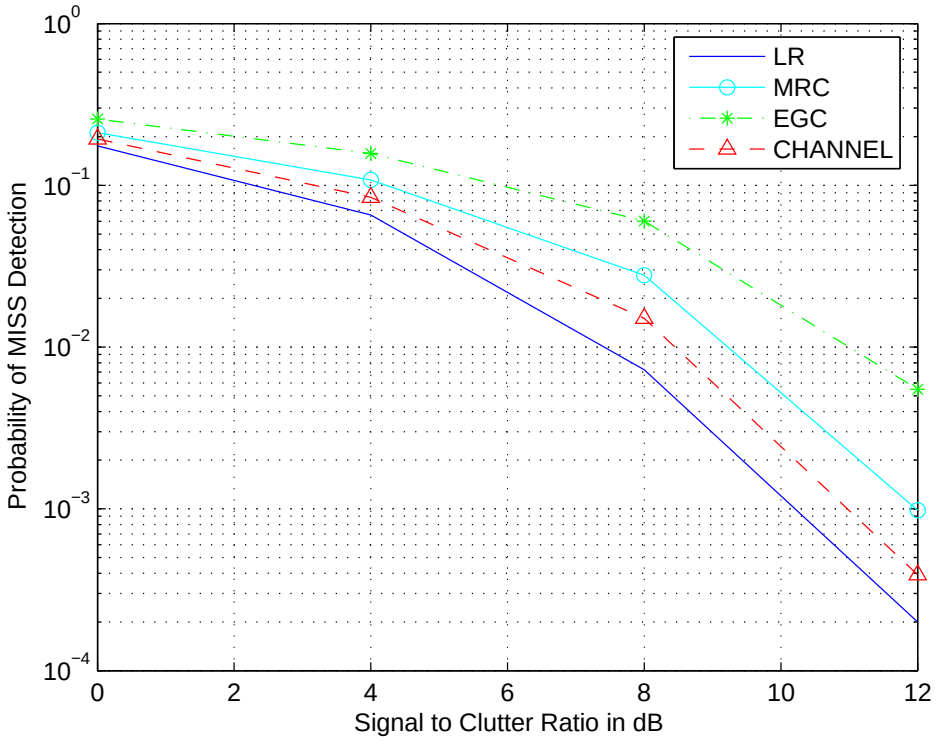


Figure 5.5. Single hop.

Fig.5.5 gives the probability of miss detection vs. the SCR for multi-target, single-hop radar sensor network. There are a total of two stationary targets and three radar sensors in the field. The optimal LR- based fusion rule provides the most powerful detection performance but it requires complete target RCS knowledge. The LR-based rule with target RCS statistics approaches the optimal LR-based rule in low SCR and have about 1dB loss in higher SCR. MRC and EGC have similar performance. Both are little worse than the LR-based rule with target RCS statistics.

Figs.5.6 and 5.7 are the performance for multi-target, multi-hop radar sensor networks. Fig.5.6 shows the probability of miss detection when each of the three radar sensors reaches the fusion center in two hops. Fig.5.7 shows the performance when the three radar sensors reach the fusion center in unequal hops. In the simulation, we assume

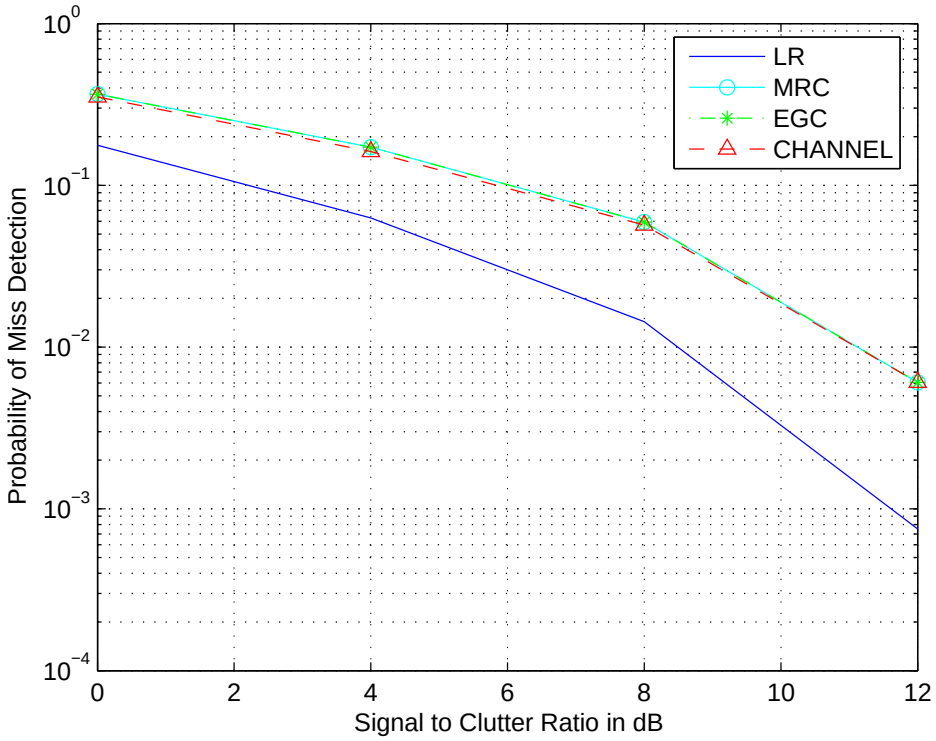


Figure 5.6. Multi-hop, equal hops.

that one radar sensor reaches the fusion center in two hops while the others in a single hop. As expected, the probability of detection for the single-hop case outperforms the one for multi-hop.

5.6 Conclusions

In this chapter, we presented the MIMO decision fusion rules for multi-target, multi-hop radar sensor networks under the assumption that the target RCS is Rayleigh model and clutter echoes follow Gaussian. We derived the optimum LR-based fusion rule and a sub-optimal LR-based fusion rule with the target RCS statistics only. Simulation results show that the *MIMO fusion* rules approach the optimal-LR and outperforms MRC and EGC at high signal to clutter ratio (SCR).

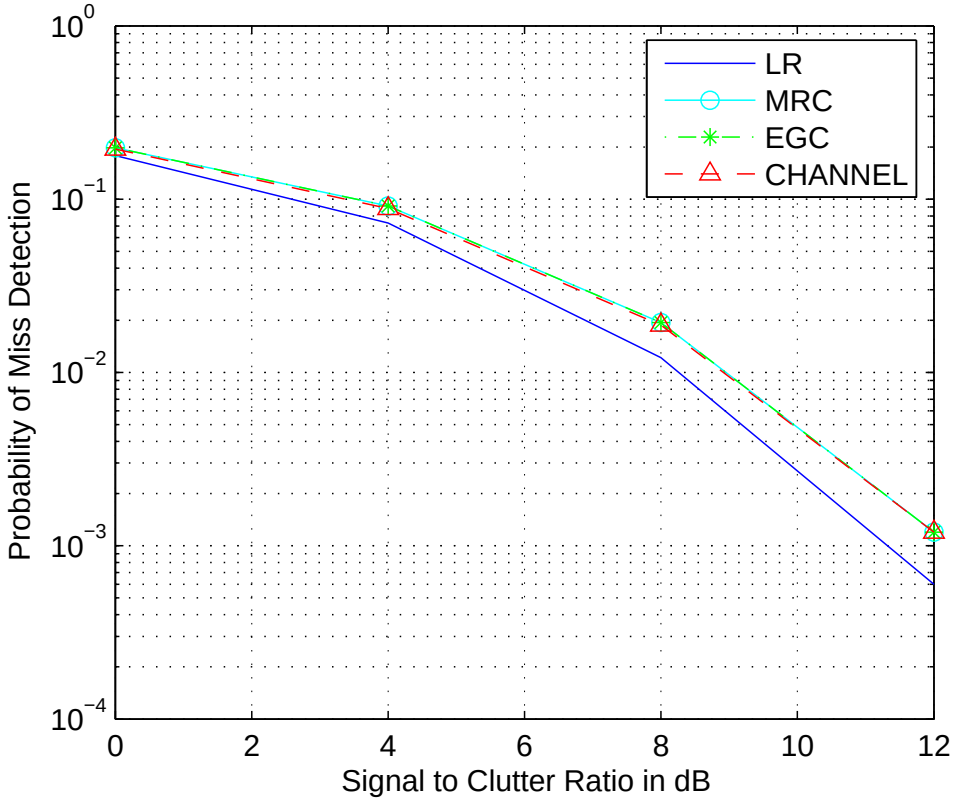


Figure 5.7. Multi-hop, unequal hops.

In many cases, two or more local radars may share a common relay node on their way to the fusion center. Under these circumstances, the independent assumption made toward the target RCS may not be held. It is actually a very interesting space correlation issue. As the radar observations always demonstrate time correlation, further research will be focused on this space-time correlation of radar sensor networks.

CHAPTER 6

SILENT POSITIONING IN UNDERWATER SENSOR NETWORKS

6.1 Introduction

UnderWater Acoustic Sensor Networks (UWA-SNs) consists of a variable number of sensors and vehicles to perform collaborative monitoring tasks over a given area. The main motivation for UWA-SNs is their relative ease of deployment since they eliminate the need for cables and they do not interfere with shipping activities. UWA-SNs are envisioned to enable applications for environmental monitoring of physical and chemical/biological indicators, tactical surveillance, disaster prevention, undersea exploration, assisted navigation, etc.

Location discovery for underwater vehicles/sensors is nontrivial in the oceanic medium. Propagation delays, motion-induced Doppler shift, phase and amplitude fluctuations, multipath interference, etc., are all significant factors in location measurement. The well-known Global Positioning System (GPS) receivers, which may be used in terrestrial systems to accurately estimate the geographical locations of sensor nodes, do not work properly in underwater [34]. Some localization schemes based on received signal strength (RSS), time of arrival (ToA), or angle of arrival (AoA), could be used. Nevertheless, the bandwidth constraint, sensor mobility, and unpredictable variation in channel behavior make many of these approaches inaccurate or unapplicable [35].

This section proposes UPS, a ToA-based silent Underwater Positioning Scheme, to carefully address the concerns and challenges mentioned above. We investigate the propagation delay and multipath channel. We found that in acoustic underwater networks with large propagation delays, a multipath channel can be modelled as a modified Ultra

Wide Band (UWB) Saleh-Valenzuela model (S-V): the arrival of each cluster and the paths within each cluster follow double Poisson distributions and the multipath channel gain follows a Rician distribution. We then analyze the theoretical performance of our scheme in propulsion noise environments and identify the possible sources of errors with measures to help mitigate them.

This section is organized as follows. Section 6.2 proposes UPS, a silent underwater positioning scheme for UWA-SN. Underwater acoustic channel modeling and theoretic performance analysis are given in Section 6.3. Simulation results are reported in Section 6.4. The conclusion is in Section 6.5.

6.2 UPS: An Underwater Positioning Scheme

In this section, we propose UPS, a silent acoustic positioning scheme for underwater vehicle/sensor localization. UPS is motivated by our previous work presented for 2D terrestrial sensor networks [61–63], which rely on the ToA of RF signals from three anchor nodes for location estimation. The propagation characteristics of RF signals in free space and that of acoustic signals underwater are significantly different, which fundamentally affect the performance of any position algorithm.

UPS consists of two steps. The first step detects the differences of signal arrival times from four anchor nodes. These time differences are transformed into range differences from the underwater vehicle/sensor to the anchor nodes. In the second step, trilateration is performed to transform these range estimates into coordinates. In the following, we first discuss the network model under our consideration.

6.2.1 Network Model

We assume that an UWA-SN consists of mobile underwater vehicles (e.g. UUVs or AUVs) and stationary sensors. UUVs and AUVs move about at a typical speed of around 2 meters [64] within a confined space, which also covers all non-mobile sensors. To ease our elaboration, from now on we use "sensor" to denote both a mobile vehicle or a stationary sensor. There exist at least four non-cospace anchor nodes with long-range beacons whose locations are known *a priori*. Each of them is equipped with an acoustic transmitter that can cover the whole activity space. No three anchors are collinear. An example layout of anchor nodes is illustrated in Fig. 6.1.

6.2.2 A Time-Based Location Detection Scheme

Given the locations (x_a, y_a, z_a), (x_b, y_b, z_b), (x_c, y_c, z_c), and (x_d, y_d, z_d) of anchor nodes A, B, C, and D, respectively, we are going to determine the location (x, y, z) of sensor S, as shown in Fig. 6.1. Let d_{ij} be the distance between i and j, where $i, j \in \{a, b, c, d, s\}$, representing the four anchor nodes and the sensor S. We have

$$\begin{aligned} d_{ab} &= \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \\ d_{ac} &= \sqrt{(x_a - x_c)^2 + (y_a - y_c)^2 + (z_a - z_c)^2} \\ d_{ad} &= \sqrt{(x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2} \end{aligned}$$

The first step of UPS computes the range differences between d_{sa} and d_{sb}, d_{sc}, d_{sd}, respectively.

Step 1: Range Difference Computation.

Let A be the master anchor node, which initiates a beacon signal every T seconds. Each beacon interval begins when A transmits a beacon signal. Consider any beacon interval i, at times t_1^i, t_b^i, t_c^i, and t_d^i, sensor S, anchor nodes B, C, and D receive A's beacon signal, respectively. At time $t_b^{i\prime}$, which is $\geq t_b^i$, B replies to A with a beacon

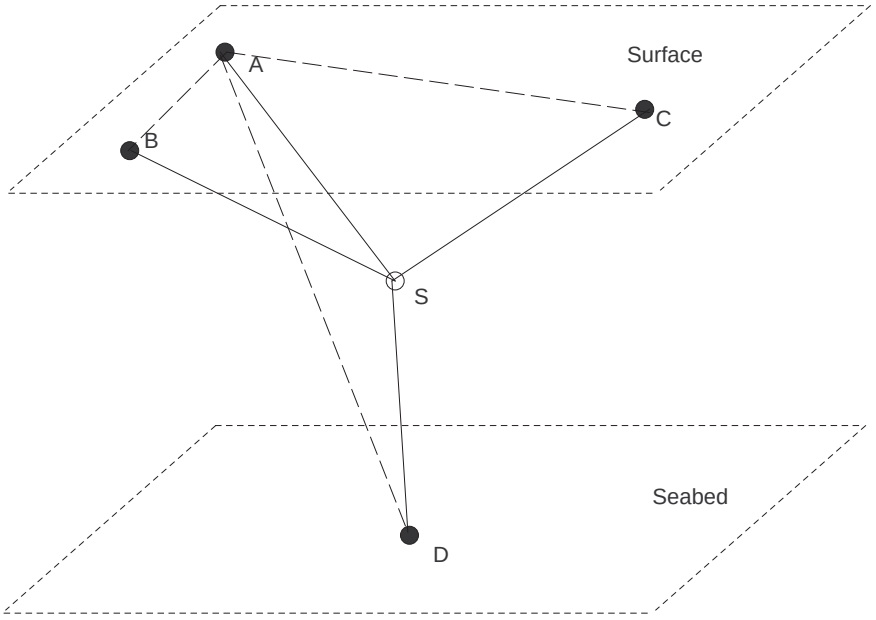


Figure 6.1. Sensor S will measure the arrival times.

signal conveying information $t_b^{i\prime} - t_b^i = \Delta t_b^i$. This signal reaches S at time t_2^i. After receiving beacon signals from both A and B, at time $t_c^{i\prime}$, C replies to A with a beacon signal conveying information $t_c^{i\prime} - t_c^i = \Delta t_c^i$. This signal reaches S at time t_3^i. After receiving beacon signals from A, B, and C, at time $t_d^{i\prime}$, D replies to A with a beacon signal conveying information $t_d^{i\prime} - t_d^i = \Delta t_d^i$. This signal reaches S at time t_4^i. Based on triangle inequality, $t_1^i < t_2^i < t_3^i < t_4^i$. Let $\Delta t_1^i = t_2^i - t_1^i$, $\Delta t_2^i = t_3^i - t_1^i$, and $\Delta t_3^i = t_4^i - t_1^i$, we obtain

$$d_{ab} + d_{sb} - d_{sa} + v \cdot \Delta t_b^i = v \cdot \Delta t_1^i, \tag{6.1}$$

$$d_{ac} + d_{sc} - d_{sa} + v \cdot \Delta t_c^i = v \cdot \Delta t_2^i, \tag{6.2}$$

$$d_{ad} + d_{sd} - d_{sa} + v \cdot \Delta t_d^i = v \cdot \Delta t_3^i, \tag{6.3}$$

which gives

$$d_{sb} = d_{sa} + v \cdot \Delta t_1^i - d_{ab} - v \cdot \Delta t_b^i = d_{sa} + k_1^i, \tag{6.4}$$

$$d_{sc} = d_{sa} + v \cdot \Delta t_2^i - d_{ac} - v \cdot \Delta t_c^i = d_{sa} + k_2^i, \tag{6.5}$$

$$d_{sd} = d_{sa} + v \cdot \Delta t_3^i - d_{ad} - v \cdot \Delta t_d^i = d_{sa} + k_3^i, \tag{6.6}$$

where d_{sa}, d_{sb}, d_{sc}, and d_{sd} are positive real numbers, v is the speed of the ultrasound, and

$$k_1^i = v \cdot \Delta t_1^i - v \cdot \Delta t_b^i - d_{ab}, \tag{6.7}$$

$$k_2^i = v \cdot \Delta t_2^i - v \cdot \Delta t_c^i - d_{ac}, \tag{6.8}$$

$$k_3^i = v \cdot \Delta t_3^i - v \cdot \Delta t_d^i - d_{ad}. \tag{6.9}$$

Averaging k_1^i, k_2^i, and k_3^i over I intervals gives

$$k_1 = \frac{v}{I}[\sum_{i=1}^{I}(\Delta t_1^i - \Delta t_b^i)] - d_{ab}, \tag{6.10}$$

$$k_2 = \frac{v}{I}[\sum_{i=1}^{I}(\Delta t_2^i - \Delta t_c^i)] - d_{ac}, \tag{6.11}$$

$$k_3 = \frac{v}{I}[\sum_{i=1}^{I}(\Delta t_3^i - \Delta t_d^i)] - d_{ad}. \tag{6.12}$$

We are going to apply trilateration with k_1, k_2, and k_3 to compute coordinates (x, y, z) for sensor S in the next step.

Remarks: (i) All arrival times, including t_j^i, where $j = 1, 2, 3, 4$, and $t_j^{i\prime}$, where $j \in \{b, c, d\}$, are based on the local timers of the anchor nodes and the sensor S. No time synchronization is required. (ii) We require A to periodically initiate the beacon signal transmission in order to decrease the measurement error and to facilitate navigation.

Step 2: Location Computation.

From Eqs. (6.4), (6.5), (6.6), (6.10), (6.11), and (6.12), we have

$$d_{sb} = d_{sa} + k_1, \tag{6.13}$$

$$d_{sc} = d_{sa} + k_2, \tag{6.14}$$

$$d_{sd} = d_{sa} + k_3. \tag{6.15}$$

Based on trilateration, we obtain four equations with four unknowns x, y, z and d_{sa}, where $d_{sa} > 0$.

$$(x - x_a)^2 + (y - y_a)^2 + (z - z_a)^2 = d_{sa}^2, \tag{6.16}$$

$$(x - x_b)^2 + (y - y_b)^2 + (z - z_b)^2 = (d_{sa} + k_1)^2, \tag{6.17}$$

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = (d_{sa} + k_2)^2, \tag{6.18}$$

$$(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2 = (d_{sa} + k_3)^2. \tag{6.19}$$

Without loss of generality, we assume that the four anchor nodes are located at $(0, 0, 0)$, $(x_b, 0, 0)$, $(x_c, y_c, 0)$, and (x_d, y_d, z_d), respectively, where $x_b > 0$, $y_c > 0$, and $z_d > 0$. Note that we can always transform real positions to this coordinate system through rotation and translation.

From Eqs. (6.16), (6.17), (6.18), and (6.19), we have

$$x^2 + y^2 + z^2 = d_{sa}^2, \tag{6.20}$$

$$(x - x_b)^2 + y^2 + z^2 = (d_{sa} + k_1)^2, \tag{6.21}$$

$$(x - x_c)^2 + (y - y_c)^2 + z^2 = (d_{sa} + k_2)^2, \tag{6.22}$$

$$(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2 = (d_{sa} + k_3)^2, \tag{6.23}$$

Solving these equations we obtain

$$d_{sa}^{(1)} = \frac{-\beta - \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \tag{6.24}$$

$$d_{sa}^{(2)} = \frac{-\beta + \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \tag{6.25}$$

$$x = A_x d_{sa} + B_y \tag{6.26}$$

$$y = A_y d_{sa} + B_y \tag{6.27}$$

$$z = A_z d_{sa} + B_z \tag{6.28}$$

where

$$\alpha = A_x^2 + A_y^2 + A_z^2 - 1 \tag{6.29}$$

$$\beta = 2(A_x B_x + A_y B_y + A_z B_z) \tag{6.30}$$

$$\gamma = B_x^2 + B_y^2 + B_z^2 \tag{6.31}$$

$$A_x = -\frac{k_1}{x_b} \tag{6.32}$$

$$B_x = \frac{x_b^2 - k_1^2}{2x_b} \tag{6.33}$$

$$A_y = \frac{k_1 x_c}{x_b y_c} - \frac{k_2}{y_c} \tag{6.34}$$

$$B_y = \frac{x_c^2 + y_c^2 - x_b x_c + \frac{x_c k_1^2}{x_b} - k_2^2}{2y_c} \tag{6.35}$$

$$A_z = \frac{k_1 x_d}{x_b z_d} - \frac{k_3}{z_d} - \frac{y_d(\frac{k_1 x_c}{x_b} - k_2)}{y_c z_d} \tag{6.36}$$

$$\begin{aligned} B_z &= \frac{x_d^2 + y_d^2 + z_d^2 - x_b x_d + \frac{x_d k_1^2}{x_b} - k_3^2 - \frac{y_d x_c^2}{y_c}}{2z_d} \\ &+ \frac{-y_c y_d + \frac{x_b x_c y_d}{y_c} - \frac{k_1^2 x_c y_d}{x_b y_c} + \frac{k_2^2 y_d}{y_c}}{2z_d} \end{aligned} \tag{6.37}$$

We have conducted extensive simulation to study the *feasible space* where $d_{sa} > 0$ is unique. It is interesting to observe that when S is not close to any anchor node, and when it is not behind any anchor node, Eq. (6.24) provides a unique feasible solution. In addition, the correct position can be computed via Eq. (6.24) if a sensor resides in the enclosed space by the four anchor nodes, even when it is close to an anchor node. Figs. 6.2-6.4 report the three transections ($z = 0, 5, 10$) of the feasible space (the gray area) when the four anchor nodes A, B, C, and D reside in $(0, 0, 0)$, $(10, 0, 0)$, $(0, 10, 0)$, and $(0, 0, 10)$, respectively.

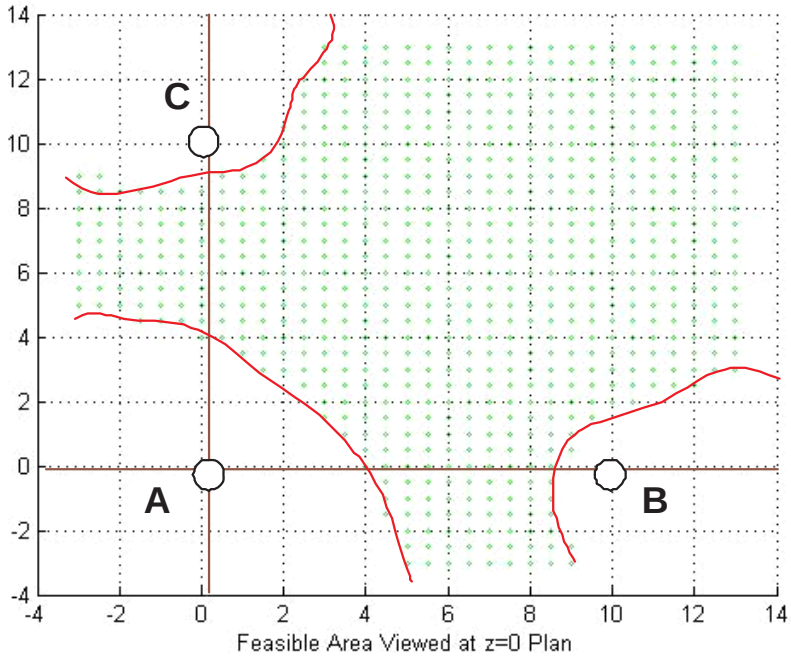


Figure 6.2. The transection of the feasible space where $z = 0$.

6.3 Channel Modeling and Theoretical Performance Analysis

6.3.1 Channel Modeling for Underwater Sensor Networks

The underwater acoustic channel exhibits phenomena such as signal fading and phase and amplitude fluctuation due to the interactions with the boundaries and the scattering from inhomogeneities within the ocean medium. The speed of sound underwater is approximately 1500 m/s, which leads to large propagation delays and motion-induced Doppler effects. Phase and amplitude fluctuations may induce high bit-error probability comparing to most radio channels. Multipath interference is another important phenomena in UWA networks, causing frequency selective fading in underwater channels.

There has been a lot of effort to model the underwater acoustic fading channels and estimate their performance. Early research [65] [66] assumes Rayleigh fading in nature

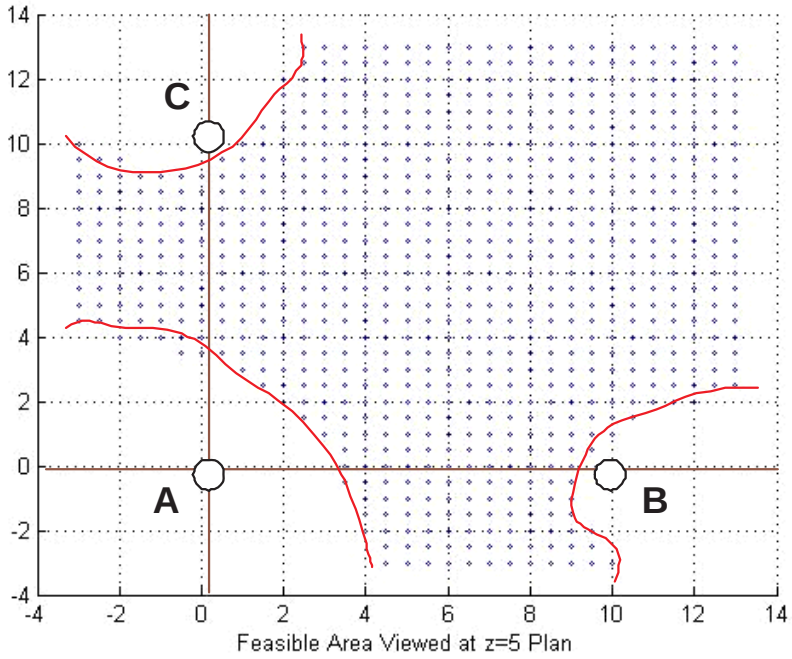


Figure 6.3. The transection of the feasible space where $z = 5$.

but later it is observed in [67] that Rayleigh fading exhibits only in limited cases. Geng and Zielinski [67] propose that in an underwater acoustic channel, there can be several distinct paths (eigenpaths) over which a signal can propagate from transmitter to receiver (eigenpath signals). Each eigenpath signal contains a dominant, stable component and many smaller, randomly scattered components (sub-eigenpath or engenray components). The envelope of the eigenpath signal can therefore be described using a Rice fading model. Such an *eigenpath* or *eigenray* concept was first introduced in [68]. Eigenray arrival angles as well as the amplitude and phase fluctuations are all modelled statistically and are assumed independent from each other. The number of eigenrays reaching the receiver is a Poisson distribution with a mean number calculated from the Ray Theory.

Enlightened by the prior research on underwater acoustic channel modeling, we propose a modified UWB Saleh-Valenzuela (S-V) channel model for underwater acoustic

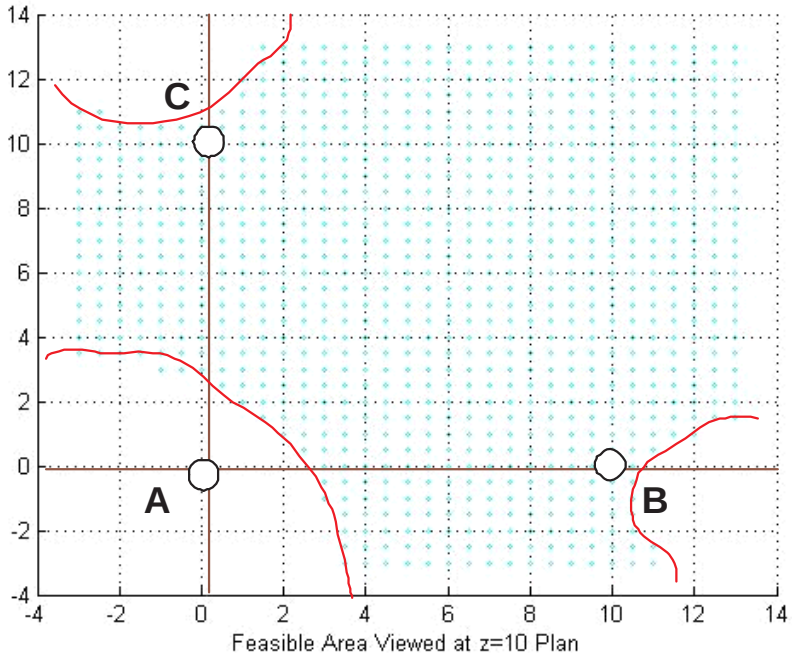


Figure 6.4. The transection of the feasible space where $z = 10$.

networks, which can be validated in three considerations. First, UWB comes from the UWB radar world and refers to the electromagnetic waveforms that are characterized by an instantaneous fractional energy bandwidth greater than about $0.20 - 0.25$. In an underwater acoustic channel, the communication frequency range is inferior to 10 kHz. In short range transmission, the carrier frequency is 550 Hz in shallow water and 2 kHz in deep water. The carrier frequency for long range transmission is 1500 Hz. In all cases, the fractional bandwidth $(f_H - f_L)/((f_H + f_L)/2)$ is greater than $0.20 - 0.25$. Therefore, the acoustic signal transmitted underwater can be classified as a UWB signal. Second, Multipath channel can be modelled as a S-V model in UWB communications. The S-V model is based on the observation that usually multipath contributions generated by the same pulse arrive at the receiver grouped into clusters. Such a channel behavior is analogous to the above eigenpath/sub-eigenpath concept in underwater networks. Third,

Similar to [68], two Poisson models are employed in the modeling of the path arrivals in UWB communications. The first Poisson model is for the first path of each path cluster and the second Poisson model is for the paths or rays within each cluster.

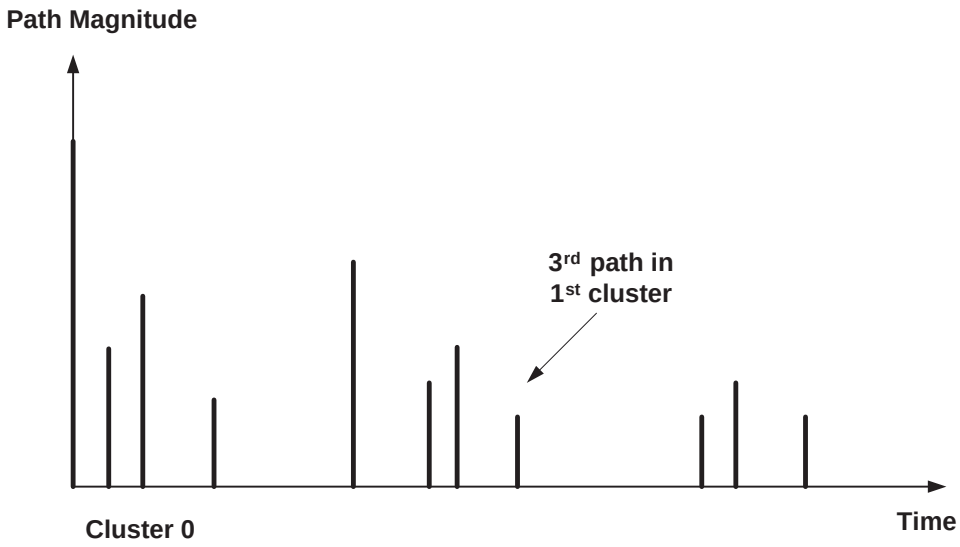


Figure 6.5. An illustration of the channel impulse response.

Applying the S-V model into underwater acoustic channels, the arrival of clusters is modelled as a Poisson arrival process with a rate Λ, while within each cluster, subsequent multipath contributions or rays also arrive according to a Poisson process with a rate λ (see Fig. 6.5). We define:

- T_l = the arrival time of the first path of the l-th cluster;
- $\tau_{k,l}$ = the delay of the k-th path within the l-th cluster relative to the first path arrival time T_l;
- Λ = the cluster arrival rate;
- λ = the ray arrival rate, i.e., the arrival rate of the paths within each cluster.

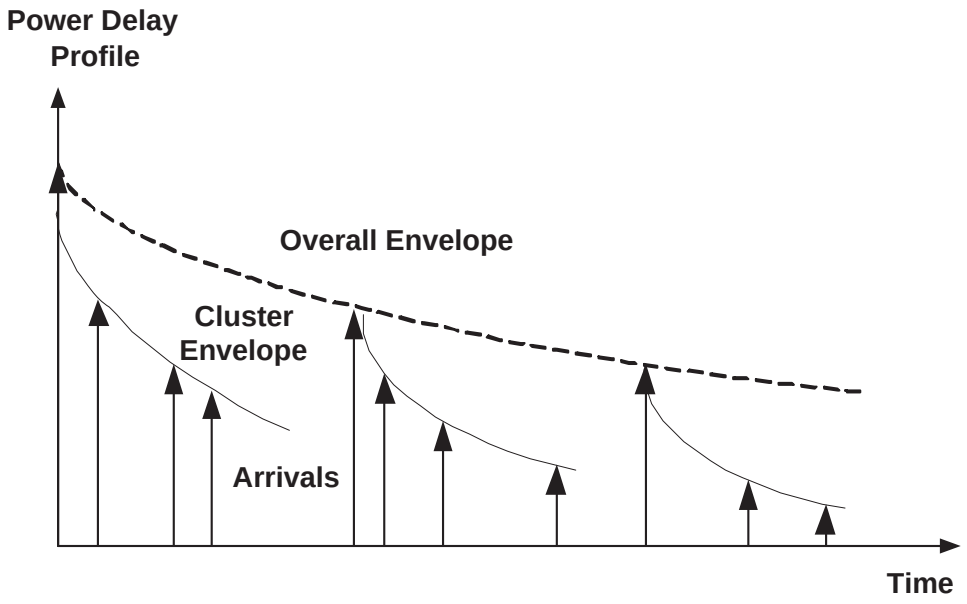


Figure 6.6. An illustration of the double exponential decay.

By definition, we have $\tau_{0l} = T_l$. The distributions of the cluster arrival time and the ray arrival time are given by

$$p(T_l|T_{l-1}) = \Lambda \exp\left(-\Lambda(T_l - T_{l-1}), l > 0\right)$$
$$p(\tau_{k,l}|\tau_{(k-1),l}) = \lambda \exp\left(-\lambda(\tau_{k,l} - \tau_{(k-1),l})\right), k > 0 \tag{6.38}$$

In the UWB S-V model, the magnitude of the k-th path within the l-th cluster is denoted by β_{kl}. It follows a Rayleigh distribution. However in underwater acoustic networks, the channel within the communication frequency range does not behave like a Rayleigh channel. Based on the discussion in [67,68], it is rather appropriate to model the multipath channel gain as a Rician distribution. Then in the underwater S-V model, the gain of the k-th path within the l-th cluster is a complex random value with a modulus β_{kl} and a phase θ_{kl}. We assume that the β_{kl} values in a underwater acoustic channel are statistically independent and are Rician distributed positive random variables, while

the θ_{kl} values are assumed to be statistically independent uniform random variables over $[0, 2\pi)$. We have

$$\overline{\beta_{kl}^2} = \overline{\beta_{00}^2} \exp\left(-T_l/\Gamma\right) \exp\left(-\tau_{kl}/\gamma\right) \tag{6.39}$$

where the term β_{00} represents the average energy of the first path of the first cluster, while Γ and γ are the power decay coefficients for clusters and multipath, respectively. According to (6.39), the average Power Decay Profile (PDP) is characterized by an exponential decay of the amplitude of the clusters, and a different exponential decay for the amplitude of the received pulses within each cluster, as shown in Fig. 6.6.

6.3.2 Theoretical Error Analysis

In this subsection, we study the position error of S resulting from the acoustic fading channel, which has been modelled as a modified UWB S-V model in Subsection 6.3.1.

The trilateration equations (6.16)-(6.19) compute the coordinates (x, y, z) for the sensor S based on the measured values k_1, k_2, and k_3, which are determined by the time-related measurements at the sensor (Δt_1^i, Δt_2^i and Δt_3^i) and anchor nodes B (Δt_b^i), C (Δt_c^i), and D (Δt_d^i) over beacon interval i (see Eqs. (6.10)-(6.12)). Therefore the errors of x, y and z result from the measuring errors of Δt_1^i, Δt_2^i, Δt_3^i, Δt_b^i, Δt_c^i, and Δt_d^i. Since the error of Δt_b^i (Δt_c^i, Δt_d^i) plays the same role as that of Δt_1^i (Δt_2^i, Δt_3^i) in the computation of k_1 (k_2, k_3), and the anchor node B (C, D) can have more sophisticated hardware to precisely estimate Δt_b^i (Δt_c^i, Δt_d^i), we consider these errors of Δt_1^i, Δt_2^i and Δt_3^i only, which are computed from t_1^i, t_2^i, t_3^i, and t_4^i, the arrival times of the beacon signals transmitted from anchor nodes A, B, C, and D at beacon internal i, respectively.

Assume the underwater sensor always listens to the first ray of the transmitted signal and records the arrival times, which in this case are t_1^i, t_2^i, t_3^i and t_4^i. Due to the underwater multi-path effect as illustrated in Fig. 6.6, these arrival times contain

an error with an exponential distribution. Let $\delta_{t1}^{\ i}$, $\delta_{t2}^{\ i}$, and $\delta_{t3}^{\ i}$ be the measuring errors of Δt_1^i, Δt_2^i, and Δt_3^i, respectively. It is reasonable to assume that $\delta_{t1}^{\ i}$, $\delta_{t2}^{\ i}$, and $\delta_{t3}^{\ i}$ are independent to each other. Given t_1^i, the conditional probability density functions (*pdf*) of $\delta_{t1}^{\ i}$, $\delta_{t2}^{\ i}$, and $\delta_{t3}^{\ i}$ are exponential with parameters λ_1, λ_2 and λ_3, respectively:

$$\begin{aligned} P_e(\delta_{t1}^{\ i}|t_1^i) &= \lambda_1 \exp(-\lambda_1 \delta_{t1}^{\ i}), \\ P_e(\delta_{t2}^{\ i}|t_1^i) &= \lambda_2 \exp(-\lambda_2 \delta_{t2}^{\ i}), \\ P_e(\delta_{t3}^{\ i}|t_1^i) &= \lambda_3 \exp(-\lambda_3 \delta_{t3}^{\ i}). \end{aligned} \tag{6.40}$$

To simplify the elaboration, we consider the case when anchor nodes A, B, C, and D are located at $(0,0,0)$, $(R,0,0)$, $(0,R,0)$, and $(0,0,R)$, respectively. To further simplify the analysis, we consider the case when S resides in a small area (with a diameter $\ll R$) whose center is equidistant to all anchor nodes. The general case can be analyzed similarly.

From Eqs. (6.10)-(6.12), k_1, k_2, and k_3 are the averaged results over I beacon intervals, and based on the Central Limit Theorem, k_1, k_2 and k_3 are approximately normally distributed when I is large. Therefore we may assume k_1, k_2, and k_3 are distributed according to $\mathcal{N}(\mu_1, \sigma_1^2)$, $\mathcal{N}(\mu_2, \sigma_2^2)$, and $\mathcal{N}(\mu_3, \sigma_3^2)$, respectively. Deducing from Eqs. (6.10)-(6.12), we have

$$\begin{aligned} k_1 : \mu_1 &= v(\frac{1}{\lambda_1} + \nu_1) - R, \qquad \sigma_1^2 = \frac{v^2}{I\lambda_1^2}, \\ k_2 : \mu_2 &= v(\frac{1}{\lambda_2} + \nu_2) - R, \qquad \sigma_2^2 = \frac{v^2}{I\lambda_2^2}, \\ k_3 : \mu_3 &= v(\frac{1}{\lambda_3} + \nu_3) - R, \qquad \sigma_3^2 = \frac{v^3}{I\lambda_3^2}. \end{aligned} \tag{6.41}$$

where ν_1, ν_2 and ν_3 are the mean of the accurate values for Δt_1^i, Δt_2^i and Δt_3^i, respectively.

Based on the assumptions, we have $\mu_1/R \approx 0$, $\mu_2/R \approx 0$ and $\mu_3/R \approx 0$. Plugging $x_b = y_c = z_d = R$ and other zero coordinates into Eq. (6.24) and simplify its solution by approximating k_1^2/R^2, k_2^2/R^2 and k_3^2/R^2 with 0, we end up with

$$d_{sa} \approx \frac{\sqrt{3R^2 + (k_1 + k_2 + k_3)^2} - (k_1 + k_2 + k_3)}{2}. \tag{6.42}$$

Substituting the above into Eq. (6.26) yields

$$x \approx \frac{R}{2} - \frac{k_1^2}{2R} - \frac{k_1}{R}\frac{\sqrt{3R^2 + (k_1 + k_2 + k_3)^2} - (k_1 + k_2 + k_3)}{2}. \tag{6.43}$$

Now replacing $\frac{k_1 k_2}{R^2} = \frac{k_1}{R}\frac{k_2}{R}$, $\frac{k_1 k_3}{R^2} = \frac{k_1}{R}\frac{k_3}{R}$ and $\frac{k_2 k_3}{R^2} = \frac{k_2}{R}\frac{k_3}{R}$ by 0, we obtain

$$x \approx \frac{R}{2} + \frac{k_1}{2R}(k_2 + k_3) - k_1\sqrt{\frac{3}{2}} = \frac{R}{2} + k_1 k_{2,3}^*, \tag{6.44}$$

where $k_{2,3}^* = (k_2 + k_3)/2R - \sqrt{3/2}$. Similarly, from Eq. (6.27) and (6.28) we have

$$\begin{aligned} y &\approx \frac{R}{2} + \frac{k_2}{2R}(k_1 + k_3) - k_2\sqrt{\frac{3}{2}} = \frac{R}{2} + k_2 k_{1,3}^*, \\ z &\approx \frac{R}{2} + \frac{k_3}{2R}(k_1 + k_2) - k_3\sqrt{\frac{3}{2}} = \frac{R}{2} + k_3 k_{1,2}^*. \end{aligned} \tag{6.45}$$

where $k_{1,3}^* = (k_1 + k_3)/2R - \sqrt{3/2}$, and $k_{1,2}^* = (k_1 + k_2)/2R - \sqrt{3/2}$.

Since (x, y, z) is used to estimate the location of S, the error in the estimation must be addressed. There are several ways to do this. The following is a common practice, where the variance of each variable is computed and the size of the variance or standard deviation is used as a measure of the estimation error.

As k_1 has a Gaussian distribution with mean μ_1 and variance σ_1^2, k_2 has a Gaussian distribution with mean μ_2 and variance σ_2^2, and k_3 has a Gaussian distribution with mean μ_3 and variance σ_3^2, the linear combination $k_{1,3}^*$ has a Gaussian distribution with mean $(\mu_1 + \mu_3)/2R - \sqrt{3/2}$ and variance $(\sigma_1^2 + \sigma_3^2)/4R^2$, $k_{1,2}^*$ has a Gaussian distribution with mean $(\mu_1 + \mu_2)/2R - \sqrt{3/2}$ and variance $(\sigma_1^2 + \sigma_2^2)/4R^2$, and $k_{2,3}^*$ has a Gaussian

distribution with mean $(\mu_2+\mu_3)/2R-\sqrt{3/2}$ and variance $(\sigma_2^2+\sigma_3^2)/4R^2$. Denote by $E(X)$ and $V(X)$ the mean and variance of a random variable X. We have, from Eq. (6.44),

$$\begin{aligned} V(x) &\approx V(k_1 k_{2,3}^*) \\ &= E(k_1 k_{2,3}^*)^2 - [E(k_1 k_{2,3}^*)]^2 \\ &= E(k_1^2 (k_{2,3}^*)^2) - [E(k_1 k_{2,3}^*)]^2. \end{aligned} \tag{6.46}$$

By the independence between k_1, k_2 and k_3, we have

$$E(k_1 k_{2,3}^*) = E(k_1)E(k_{2,3}^*) \tag{6.47}$$

$$\begin{aligned} E(k_1^2 (k_{2,3}^*)^2) &= E(k_1^2)E(k_{2,3}{*}^2) \\ &= [V(k_1) + (E(k_1))^2] \\ &\quad [V(k_{2,3}^*) + (E(k_{2,3}^*))^2]. \end{aligned} \tag{6.48}$$

Therefore substitution gives

$$\begin{aligned} V(x) &\approx V(k_1)[E(k_{2,3}^*)]^2 + V(k_{2,3}^*)[E(k_1)]^2 \\ &\quad + V(k_1)V(k_{2,3}^*) \\ &= \sigma_1^2 \left(\frac{\mu_2+\mu_3}{2R} - \sqrt{\frac{3}{2}} \right)^2 + \frac{\sigma_2^2+\sigma_3^2}{4R^2}\mu_1^2 \\ &\quad + \sigma_1^2 \frac{\sigma_2^2+\sigma_3^2}{4R^2} \\ &= \frac{\sigma_1^2(\mu_2+\mu_3)^2 + (\sigma_2^2+\sigma_3^2)\mu_1^2 + \sigma_1^2(\sigma_2^2+\sigma_3^2)}{4R^2} \\ &\quad + \sigma_1^2 \left(\frac{3}{2} - \frac{\mu_2+\mu_3}{R}\sqrt{\frac{3}{2}} \right). \end{aligned} \tag{6.49}$$

Since $\mu_1/R \approx 0$, $\mu_2/R \approx 0$, plugging in Eq.(6.42), the above reduces to

$$\begin{aligned} V(x) &\approx \frac{\sigma_1^2}{2}\left(3 + \frac{\sigma_2^2+\sigma_3^2}{2R^2}\right) \\ &= \frac{v^2}{2I\lambda_1^2}\left[3 + \frac{v^2}{2IR(\lambda_2^2+\lambda_3^2)}\right]. \end{aligned} \tag{6.50}$$

Similarly, we have

$$\begin{aligned} V(y) &\approx V(k_{1,3}^*)[E(k_2)]^2 + V(k_2)[E(k_{1,3}^*)]^2 \\ &\quad + V(k_{1,3}^*)V(k_2) \\ &\approx \frac{\sigma_2^2}{2}\left(3 + \frac{\sigma_1^2+\sigma_3^2}{2R^2}\right) \\ &= \frac{v^2}{2I\lambda_2^2}\left[3 + \frac{v^2}{2IR(\lambda_1^2+\lambda_3^2)}\right]. \end{aligned} \tag{6.51}$$

and

$$\begin{aligned} V(z) &\approx V(k_{1,2}^*)[E(k_3)]^2 + V(k_3)[E(k_{1,2}^*)]^2 \\ &\quad + V(k_{1,2}^*)V(k_3) \\ &\approx \frac{\sigma_3^2}{2}\left(3 + \frac{\sigma_1^2+\sigma_2^2}{2R^2}\right) \\ &= \frac{v^2}{2I\lambda_3^2}\left[3 + \frac{v^2}{2IR(\lambda_1^2+\lambda_2^2)}\right]. \end{aligned} \tag{6.52}$$

From the above analysis, we have the following observations. First, the variances of x, y and z depend on the ray arrival rates λ_1, λ_2 and λ_3. Second, λ_1 contributes more to the variance of x than λ_2 and λ_3, λ_2 contributes more to the variance of y than λ_1 and λ_3, and λ_3 contributes more to the variance of z than λ_1 and λ_2. Third, when R is large, $V(x) \approx 3\sigma_1^2/2 = 3v^2/2I\lambda_1^2$, $V(y) \approx 3\sigma_2^2/2 = 3v^2/2I\lambda_2^2$, and $V(z) \approx 3\sigma_3^2/2 = 3v^2/2I\lambda_3^2$, showing that the variance of x is dependent on that of k_1, the variance of y is dependent on that of k_2, and the variance of z is dependent on that of k_3. Fourth, if $\lambda_1 = \lambda_2 = \lambda_3$, the variances of x, y and z can be treated the same in practice.

Note that the above analysis well-explain our simulation results, which indicate position errors strongly depend on the arrival rates of double exponential distributions.

6.3.3 Sources of Errors

There are three major sources of errors for time-based location detection schemes in UWA-SNs: *the receiver system delay*, *the underwater multipath fading*, and *the variable acoustic speed underwater*. The receiver system delay is the time duration from which the signal hits the receiver antenna until the signal is decoded accurately by the receiver. This time delay is determined by the receiver electronics. Usually it is constant or varies in very small scale when the receiver and the channel are free from interference. This system delay can be predetermined and be used to calibrate the measurements. For example, anchor nodes B, C, and D can always eliminate the system delay from Δt_b^i, Δt_c^i and Δt_d^i before these values are conveyed to the sensors in their reply messages to A's beacon signal. Meanwhile, as Δt_1^i, Δt_2^i and Δt_3^2 are measured by one sensor, the effect of receiver system delay may cancel out. Thus in our model, if anchor nodes B, C, and D can provide precise *a priori* information on receiver system time delays, the effect of these delays will be negligible.

The underwater multipath fading channel will greatly influence the location accuracy of any location detection system. Major factors influencing terrestrial multipath fading [**?**] include *multipath propagation*, *speed of the receiver*, *speed of the surrounding objects*, and the *transmission signal bandwidth*. In the underwater environment, other important factors include *water temperature and clarity*, *motion behavior of receiver and underwater objects and transmission range*. In our time-based location scheme, we assume that the motion of the underwater vehicles is relatively small such that the motion-induced Doppler effect can be ignored.

There are two important characteristics of multipath signals. First, the multiple non-direct path signals will always arrive at the receiver antennae latter than the direct path signal, as they must travel a longer distance. Second, in LOS transmission model, non-direct multipath signals will normally be weaker than the direct path signal, as

some signal power will be lost from scattering. If NLOS exists, the non-direct multipath signal may be stronger, as the direct path is hindered in some way. Based on these characteristics, scientists can always design more sensitive receivers to lock and track the direct path signal. For example, multipath signals using a pseudo-random code arriving at the receiver later than the direct path signal will have negligible effects on a high-resolution DS-BPSK receiver [69]. Our location detection scheme mitigates the effect of multipath fading by measuring TDoA over multiple beacon intervals and modeling the multipath arrival times as the double exponential distribution. TDoA measurements have been very effective in fading channels, as many detrimental effects caused by multipath fading and processing delay can be cancelled [70].

Another source of position error is the variable speed of sound, which significantly affects the precision of all localization systems that assume a constant acoustic speed. The velocity of underwater acoustics depends on temperature, salinity, and depth [71]. As a future research, we will investigate the influence of the variable speed of sound on our positioning scheme.

6.4 Simulation

In this section, we are going to study the performance of UPS in UWA-SN. We take the same assumptions as those in Subsection 6.3.2, that for a given t_1^i, the measuring errors of Δt_i^i, Δt_2^i and Δt_3^i are independent exponential distributions with arrival rates λ_1, λ_2 and λ_3, respectively, and that the underwater vehicle/sensor measures the arrival time of the first ray of the first cluster only in a multipath fading channel. We further assume that the measuring errors of Δt_b^i, Δt_c^i and Δt_d^i are negligible, as justified in Subsection 6.3.3. Therefore λ_1, λ_2 and λ_3 can be assumed to be equal, and this value is denoted by λ. We will investigate the influence of λ upon position error. Another

factor we will investigate is the number of beacon intervals I used to compute k_1, k_2, and k_3. Since we also consider the localization of mobile underwater vehicles, we choose to average over a small number of beacon intervals.

We use Matlab to code UPS. This tool provides procedures to generate normally distributed and uniformly distributed random numbers. Note that we do not use the *sqrt* function in Matlab. Instead, we use Newton's method [72]. We have found that 4 iterations generally yield good results. In addition, Eq. (6.24) will be adopted for position estimation since in our simulation, sensors will be placed within the space enclosed by four anchor nodes.

First we study the distribution of position errors over a 3D monitored space. In this simulation scenario, the four anchor nodes are located at $(0, 0, 0)$, $(20, 0, 0)$, $(0, 20, 0)$, and $(0, 0, 20)$ respectively. Sensors are placed at grid points $(i + 0.5, j + 0.5, k + 0.5)$, where $i, j, k = 0, 1, \cdots, 19$. The errors of Δt_1^i, Δt_2^i and Δt_3^i are exponentially distributed with parameters $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$. We average the sensor location estimation over 1000 trials. For each trial, $I = 16$. In addition, we simulate different λ settings and obtain similar results. Nevertheless, we report the cases of $\lambda = 0.1$ and $\lambda = 2.5$ only in this paper. Fig. 6.7 and Fig. 6.8 illustrate the $(x - y)$ plane position errors vs. the real positions of the sensors for $z = 5$. Results from $(x - z)$ and $(y - z)$ planes are close to those reported for the $(x - y)$ plane.

We observe that a lower arrival rate gives a better estimation since a higher arrival rate may bring motion-induced Doppler shift in the channel and cause jittering in the measurement. We also observe that as the distances from a sensor to all four anchor nodes become larger, the position errors will become larger correspondingly; when the sensor is closer to any of the four anchor nodes, the errors become larger. Notice that in Fig. 6.7 the sensor at location $(9.5, 9.5, 5.0)$, which is close to the intersection of the three angle bisectors of $\triangle ABC$, has the smallest position error and the sensors at its neighboring

area also demonstrate quite low position errors. Interestingly, Refs. [61, 73, 74] provide similar results in their simulation study. Intuitively this is because the geometry of the intersection of the range circles is poor when the sensors are far away from any anchor node or when the sensors are close to any anchor node. From this analysis, we conclude that the position error is related to the placement of anchor nodes. Careful studies will be conducted in the future as the results can be applied to guide the deployment of anchor nodes for better performance. For these reasons, in the following simulation, we intentionally enforce an allowable shortest distance (1.0 unit) from any randomly generated sensor to any anchor node. This means the four anchor nodes are placed some distance away from the boundary of the monitored area.

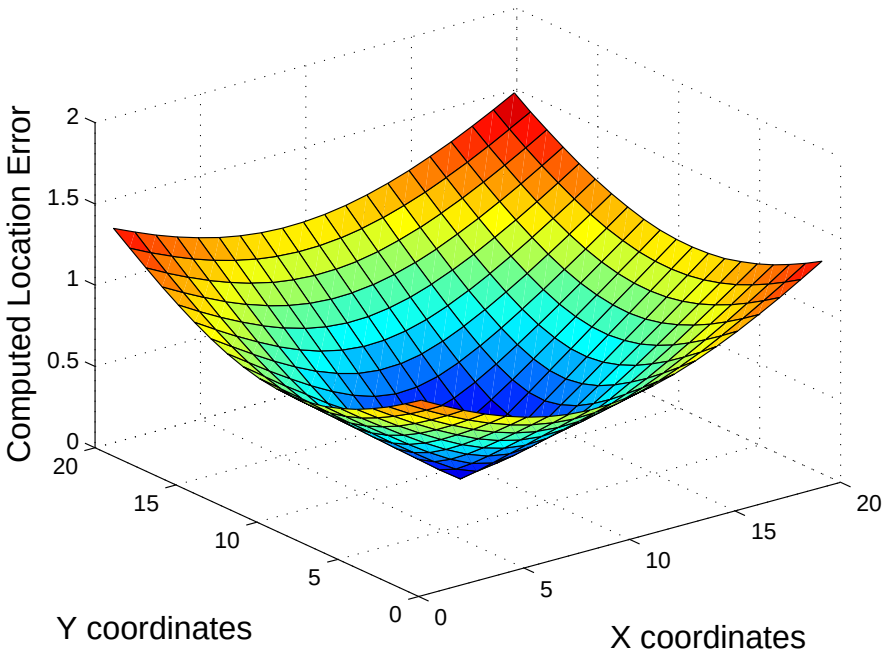


Figure 6.7. Position errors *vs.* real positions when $\lambda = 0.1$ in $z = 5$ plane .

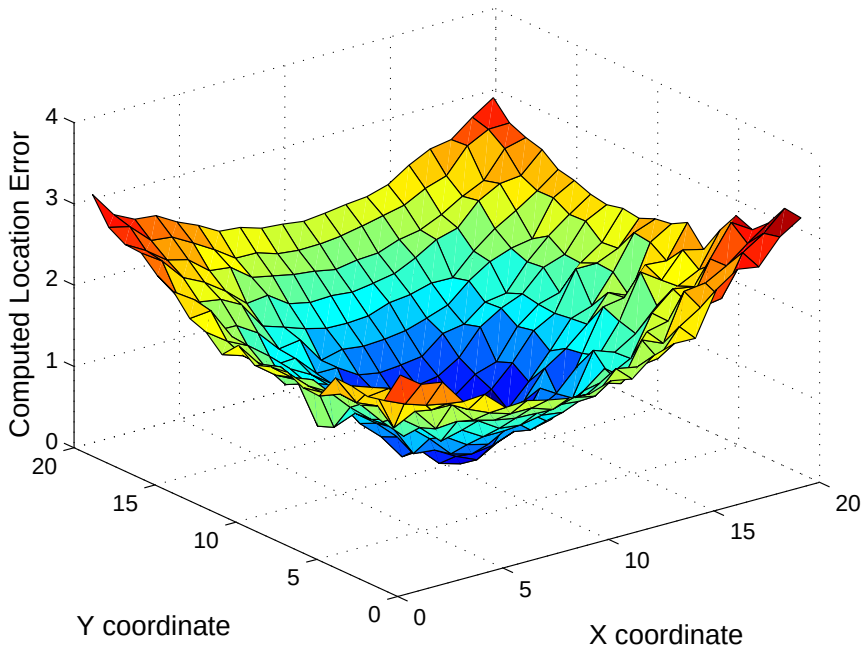


Figure 6.8. Position errors *vs.* real positions when $\lambda = 2.5$ in $z = 10$ plane.

Next we consider the scenario when sensors are randomly deployed in a cubic space with lower-left corner $(1, 1, 1)$ and upper-right corner $(19, 19, 19)$. The four anchor nodes are still located at $(0, 0, 0)$, $(20, 0, 0)$, $(0, 20, 0)$ and $(0, 0, 20)$, respectively. For each λ value, we try 2000 random sensor positions. The averaged results are reported in Fig. 6.9. Note that in this study, I is selected from $\{1, 2, 3, 4\}$ in order to demonstrate the effectiveness of UPS when applied to positioning mobile underwater vehicles.

We obtain three observations from Fig. 6.9. First, as I increases, position error decreases. This is because averaging over larger number of beacon intervals to compute k_1 k_2 and k_3 can better smooth out the effects of measuring errors in Δt_1^i, Δt_2^i and Δt_3^i, thus produce improved result. A detailed theoretical explanation comes from Subsection 6.3.2. As I increases, σ_1^2, σ_2^2 and σ_3^2 will decrease, and thus $V(x)$, $V(y)$ and $V(z)$ will decrease. Then the errors from estimating the coordinates of sensors by x, y and z will decrease,

implying that the position error will become smaller. Second, position error increases as λ increases. This is reasonable in the underwater acoustic channel, in which a higher λ comes from an even higher transmission rate when asymmetry commonly exists between the transmitter and the receiver. Such characteristic of the underwater medium brings significant multipath interference at the receiver and causes jittering as illustrated in Fig. 6.8. Again, this can be well explained by Subsection 6.3.2. In fact, if λ increases which means v increases at an even larger pace, v^2/λ^2 increases. As a result, $V(x)$, $V(y)$ and $V(z)$ increase so that the errors from estimating the coordinates of the sensor by x and y and z increase. Thus the larger the arrival rate, the larger the position error. Third, in the situation of small λ, for example $\lambda \approx 0.5$ as shown in Fig. 6.9, the location errors vary very little with the number of beacon intervals I. When λ is relatively high, I plays a more important role. The higher the λ, the bigger the difference induced from I. This observation is analogous to the terrestrial wireless communication channels, in which *coherenct time* is introduced to depict a period of time where the channel behavior or model can be considered as stationary. For underwater wireless communications, not only temporal coherence but also spatial and frequency coherences [68] are significant parameters for signal propagation through acoustic channels with multiple paths. Based on the third observation, λ in underwater communications should not be neglected in estimating the coherence parameters. Note that rotating the square-cube monitored space within the open space formed by anchor nodes A, B, C and D, we obtain very similar results.

In the following, we report the simulation results when $\angle BAC \leq 90°$. In this simulation, the four anchor nodes are located at $(0, 0, 0)$, $(X_B, Y_B, 0)$, $(X_C, Y_C, 0)$, and $(0, 0, Z_D)$, respectively, where X_B, Y_B, X_C, Y_C and Z_D are randomly drawn from $[5, 20]$. 2000 sensors are randomly placed within the overlapping space formed by the anchor nodes (A, B, C, D) and the cube space with corners $(0, 0, 0)$ and $(20, 20, 20)$. Fig. 6.10

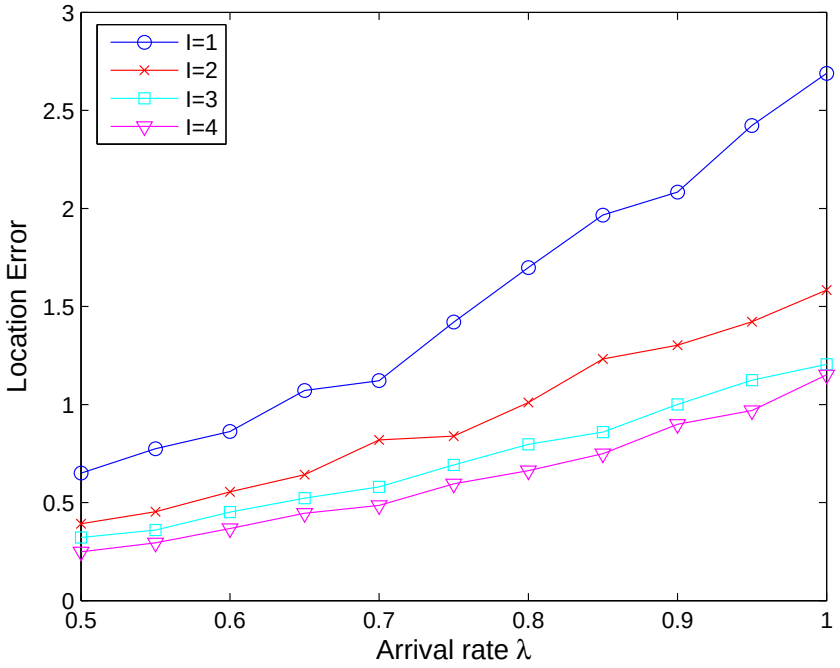


Figure 6.9. Position errors *vs.* λ where $\angle BAC = 90^\circ$.

reports the position error *vs.* λ. Note that the observations from Fig. 6.10 is very similar to those from Fig. 6.9. Nevertheless, for the same λ, the acute angle case performs slightly better than the right angle one.

6.5 Conclusion

In this section, we propose UPS, a silent underwater positioning scheme for UWA-SNs. UPS is superior to existing systems in many aspects such as synchronization-free, low computation overhead, etc. To evaluate the performance of UPS, we model the underwater acoustic channel with a modified UWB S-V model, and conduct both theoretical analysis and simulation study. Our scheme is simple and effective.

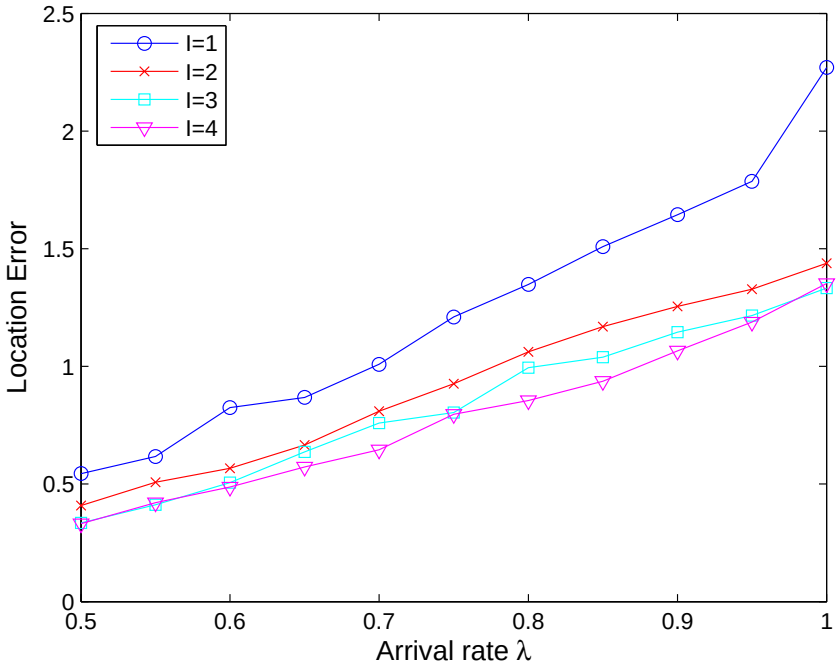


Figure 6.10. Position errors vs. λ where $\angle BAC \leq 90°$.

CHAPTER 7

CONCLUSIONS

This chapter concludes the dissertation. The chapter begins with a summary of the dissertation work followed with a discussion of future research directions in the energy efficient wireless sensor network technologies.

7.1 Summary of Research Work

In this dissertation report, we described a new method based on fuzzy logic systems to analyze and estimate the network lifetime for wireless sensor networks. Our approach is illuminated by the research that a single node lifetime behaves the nature of normal Gaussian distribution. However,we showed that if the single node lifetime follows normal Gaussian distribution, it is most appropriate to be modeled as a Gaussian MF with uncertain standard deviation. We then set up the Fuzzy Logic Lifetime Evaluator (FLLE) based on interval type-2 FLSs for lifetime estimation and test its performance using real lifetime data. Simulation results convincingly justified the feasibility of applying type-2 FLSs into wireless sensor network lifetime analysis. We believe that our approaches opens up a new vision for research on wireless sensor network lifetime analysis. (Chap 2)

For the sensor deployment topic, we proposed a sensor deployment strategy based on fuzzy logic system. Our approach has a great advantage to deal with the uncertainty in distributed sensor deployment which is particularly useful when emergency rescue or redeployment over hostile situation is needed. In an energy constrained wireless sensor network, fast and efficient deployment strategy is a necessity to save energy and extend network lifetime. Our FLSs scheme is capable to model all distributed sensor deployment

with a fuzzy logic system. The network coverage and quality of communication in term of outage probability are greatly improved as a result. Moreover, the FLSs scheme brings the whole network to a stable and optimal deployment very soon which will significantly reduce the energy consumption. (Chap 3)

In the case of distributed source coding, we have proposed a spectrum efficient coding scheme for correlated non-binary sources in sensor networks. Instead of using theoretically ideal data, our scheme is based on the statistic characters of the correlated non-binary sources from real sensor network. The coset construction introduced in this paper leverages the inherent correlations between sensor observations, but more importantly by minimizing the cross ratio, decreases the probability of decoding error. The proposed scheme performs at 0.5 - 1.5 dB from the Wyner-Ziv distortion bound. Our approach provides a practical solution to distributedly compress the acoustic sensor observations and can be extended to the CEO problem. (Chap 4)

we presented the MIMO decision fusion rules for multi-target, multi-hop radar sensor networks under the assumption that the target RCS is Rayleigh model and clutter echoes follow Gaussian. We derived the optimum LR-based fusion rule and a sub-optimal LR-based fusion rule with the target RCS statistics only. Simulation results show that the *MIMO fusion* rules approach the optimal-LR and outperforms MRC and EGC at high signal to clutter ratio (SCR). (Chap 5)

Considering the underwater acoustic sensor application, we propose UPS, a silent underwater positioning scheme for UWA-SNs. UPS is superior to existing systems in many aspects such as synchronization-free, low computation overhead,etc. To evaluate the performance of UPS, we model the underwater acoustic channel with a modified UWB S-V model, and conduct both theoretical analysis and simulation study. Our scheme is simple and effective. (Chap 6)

7.2 Future Work

In previous chapters we have discussed a new approach to analyze the lifetime of a wireless sensor network based on interval type-2 fuzzy logic systems and two energy efficient techniques in wireless sensor networks: fuzzy optimization for distributed sensor deployment, spectrum efficient coding scheme for correlated non-binary sources in wireless sensor networks and two sensor applications: data fusion in a multi-target radar sensor networks and silent positioning in underwater acoustic sensor networks. These topics can be extended to more meaningful collaborative positioning and tracking with multiple sensors.

A sensor network is designed to perform a set of high-level information processing tasks such as detection, positioning tracking, or classification. Energy-constrained networked sensing systems rely on collaborative signal and information processing (CSIP) to dynamically allocate resources based on task requirements. Target positioning and tracking is used as a canonical problem for studying CSIP issue. It is especially useful for illustrating a central problem of CSIP: dynamically defining and forming sensor groups based on task requirements and resource availability. In this section, I propose my future works on target positioning and tracking in radar or underwater sensor systems.

When multiple sensors view a common area, there can be improved tracking, the data rate can be greater than any of the sensors acting alone, there is less vulnerability to electronic countermeasures, and less likelihood of having missed detections due to reduced echo-signal strength caused by nulls in one of the antenna patterns or changes in the target aspect. There are two related cases: one is when the sensors are collocated, as on the same ship or at the same land site; the other, is when they are at separated locations and netted together. The first case will be studied in future research. We combine all the detections from each radar to form a single track and to update the track rather than develope separate tracks at each sensor and either select the best track or

combine them in some other manner. The development of a single track file by the use of the total data available from all sensors produces a better track than combining the tracks developed individually at each sensor. It reduces the likelihood of a loss of data as might be caused by antenna lobing, target fading, interference, and clutter since intergrated processing permits the favorable weighting of the better data and lesser weighting of the poorer data. One interesting question in tracking maneuvering targets with multiple sensors is: does more data always mean better estimates? When the problem of tracking maneuvering targets with multiple sensors is considered in the literature, the number and type of sensors that support a given target track is usually fixed with respect to a given location of the target. However, in many multisensor systems, the number and type of sensors supporting a particular target track can vary with time due to the mobility, type and resource limitations of the individual sensors. This variability in the configuration of the sensor system poses a significant problem when tracking maneuvering targets because of the uncertainty in the target motion model. [75] has shown that more data does not necessarily mean better target state estimate. More data means better performance only if the track filter is consistent. Results in [75] has to satisfy certain criteria and I am interested to learn if there is a universal condition under which more data does not necessarily mean better performance, moreover how to associate the observed data, reducing the redundancy to achieve better target state estimate.

APPENDIX A

MULTIPLE LOG-NORMAL INTERFERERS

Consider the sum of N_I log-normal random variables

$$I = \sum_{k=1}^{N_I} \Omega_k = \sum_{k=1}^{N_I} 10^{\Omega_{k(dBm)}/10} \tag{A.1}$$

where the $\Omega_{k(dBm)}$ are Gaussian random variables with mean $\mu_{\Omega_{k(dBm)}}$ and variance $\sigma^2_{\Omega_k}$, and the $\Omega_k = 10^{\Omega_{k(dBm)}/10}$ are the log-normal random variables. Unfortunately, there is no known closed form expression for the probability density function (pdf) of the sum of multiple ($N_I \geq 2$) log-normal random variables. However, there is a general consensus that the sum of independent log-normal random variables can be approximated by another log-normal random variable with appropriately chosen parameters. That is,

$$I = \sum_{k=1}^{N_I} 10^{\Omega_{k(dBm)}/10} \approx 10^{Z_{(dBm)}/10} = \hat{I} \tag{A.2}$$

where $Z_{(dBm)}$ is a Gaussian random variable with mean $\mu_{Z(dBm)}$ and variance σ^2_Z. The problem is to determine $\mu_{Z(dBm)}$ and variance σ^2_Z in terms of the $\mu_{\Omega_{k(dBm)}}$ and variance $\sigma^2_{\Omega_k}$, $k = 1, ..., N_I$. Several methods have been suggested in the literature to solve this problem including those by Fenton, Schwartz and Yen, and Farley. Each of these methods provides varying degrees of accuracy over specified ranges of the shadow standard deviation σ_Ω, the sum I, and the number of interferes N_I.

APPENDIX B

FENTON-WILKINSON METHOD

With the Fenton-Wilkinson method, the mean $\mu_{Z(dBm)}$ and variance σ_Z^2 of $Z_{(dBm)}$ are obtained by matching the first two moments of the sum I with the first two moments of the approximation $\hat{I}$. To derive the appropriate moments, it is convenient to use natural logarithms. We write

$$\Omega_k = 10^{\Omega_{k(dBm)}/10} = e^{\epsilon\Omega_{k(dBm)}} = e^{\hat{\Omega}_k} \tag{B.1}$$

where $\epsilon = (ln10)/10 = 0.23026$ and $\hat{\Omega}_k = \epsilon\Omega_{k(dBm)}$. Note that $\mu_{\hat{\Omega}_k} = \epsilon\mu_{\Omega_{k(dBm)}}$ and $\sigma^2_{\hat{\Omega}_k} = \epsilon^2\sigma^2_{\Omega_k}$. The nth moment of the log-normal random variable Ω_k can be obtained from the moment generating function of the Gaussian random variables $\hat{\Omega}_k$ as

$$E[\Omega_k^n] = E[e^{n\hat{\Omega}_k}] = e^{n\mu_{\hat{\Omega}_k} + (1/2)n^2\sigma^2_{\hat{\Omega}_k}} \tag{B.2}$$

To find the appropriate moments for the log-normal approximation we can use (14) and equate the first two moments on both sides of the equation

$$I = \sum_{k=1}^{N_I} e^{\hat{\Omega}_k} \approx e^{\hat{Z}} = \hat{I} \tag{B.3}$$

where $\hat{Z} = \epsilon Z_{(dBm)}$. For example, suppose that $\hat{\Omega}_k$, $k = 1, ..., N_I$ have mean $\mu_{\hat{\Omega}_k}$, $k = 1, ..., N_I$ and identical variances $\sigma^2_{\hat{\Omega}}$. Identical variances are often assumed because the standard deviation of log-normal shadowing is largely independent of the radio path length. Equating the means on both sides of (15)

$$\mu_I = E[I] = \sum_{k=1}^{N_I} E[e^{\hat{\Omega}_k}] = E[e^{\hat{Z}}] = E[\hat{I}] = \mu_{\hat{I}} \tag{B.4}$$

gives the result

$$\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}} \right) e^{(1/2)\sigma^2_{\hat{\Omega}}} = e^{\mu_{\hat{Z}} + (1/2)\sigma^2_{\hat{Z}}} \tag{B.5}$$

Likewise, we can equate the variances on both sides of (15) under the assumption that the $\hat{\Omega}_k$, $k = 1, ..., N_I$ are independent

$$\sigma_I^2 = E[I^2] - \mu_I^2 = E[\hat{I}^2] = \sigma_{\hat{I}}^2 \tag{B.6}$$

giving the result

$$\left(\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}} \right) e^{\sigma_{\hat{\Omega}}^2} (e^{\sigma_{\hat{\Omega}}^2} - 1) = e^{2\mu_{\hat{Z}}} e^{\sigma_{\hat{Z}}^2} (e^{\sigma_{\hat{Z}}^2} - 1) \tag{B.7}$$

By squaring each side of (17) and dividing each side of resulting equation by the respective side of (19) We can solve for $\sigma_{hat_Z}^2$ in terms of the known values of $\mu_{\hat{\Omega}_k}$, $k = 1, ..., N_I$ and $\sigma_{\hat{\Omega}}^2$. Afterwards, $\mu_{\hat{Z}}$ can be obtained from (17). This procedure yields the following solution:

$$\mu_{\hat{Z}} = \frac{\sigma_{\hat{\Omega}}^2 - \sigma_{\hat{Z}}^2}{2} + ln \left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}} \right) \tag{B.8}$$

$$\sigma_{\hat{Z}}^2 = ln \left((e^{\sigma_{\hat{\Omega}}^2} - 1) \frac{\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}}}{(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}})^2} + 1 \right) \tag{B.9}$$

Finally, $\mu_{Z(dBm)} = \epsilon^{-1} \mu_{\hat{Z}}$ and $\sigma_Z^2 = \epsilon^{-2} \sigma_{\hat{Z}}^2$.

The accuracy of this log-normal approximation can be measured in terms of how accurately the first two moments of $I_{(dB)} = 10 log_{10} I$ are estimated, and how well the cumulative distribution function (cdf) of $I_{(dB)}$ is described by a Gaussian cdf. In problems relating to the co-channel interference outage in cellular radio systems, we are usually interested in the tails of both the complementary distribution function (cdfc) $F_I^C = P(I \geq x)$ and the cdf $F_I(x) = 1 - F_I^C = P(I < x)$. In this case, we are interested in the accuracy of the approximation

$$F_I(x) \approx P(e^{\hat{Z}} \geq x) = Q \left(\frac{lnx - \mu_{\hat{Z}}}{\sigma_{\hat{Z}}} \right) \tag{B.10}$$

for large and small values of x. It will be shown later that the Fenton-Wilkinson method can approximate the tails of the cdf and cdfc functions with good accuracy.

REFERENCES

[1] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," in *Proc. IEEE*, vol. 91, no. 8, Aug. 2003, pp. 1247–1256.

[2] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer, and T. W. Kenny, "Two-tired wireless sensor network architecture for structural health monitoring," in *Proc. SPIE*, San Diego, CA, Mar. 2003.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. WSNA*, Atlanta, Georgia, Sept. 2002.

[4] M. Bhardwaj, T. Garnett, and A. Chandrakasan, "Upper bounds on the lifetime of sensor networks," in *Proc. IEEE International Conference on Communications*, 2001, pp. 785–790.

[5] M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. INFOCOM*, 2002, pp. 1587–1596.

[6] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hod networks," in *Proc. MOBICOM*, Atlanta, Georgia, Sept. 2002.

[7] E. Jain and Q. Liang, *Sensor placement and lifetime of wireless sensor networks: theory and performance analysis*, ser. Sensor Network Operations, S. Phoha, T. F. L. Porta, and C. Griffin, Eds. John Wiley & Sons, 2005.

[8] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes(discus):design and construction," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, Mar. 2003.

[9] ——, "Distributed source coding:symmetric rates and applications to sensor networks," in *Proc. DCC'00.*

[10] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2.

[11] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," in *IEEE International Conference on Wireless Communications and Networking*, vol. 3.

[12] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *IEEE International Conference on Robotics and Automation*, vol. 1.

[13] E. Biagioni and G.Sasaki, "Wireless sensor placement for reliable and efficient data collection," in *Proc. Hawaii Int.Conf.Syst.Sci.*

[14] C. Y. Chong and S. P. . Kumar, "Sensor networks: Evolution, opportunities, and challenges," in *Proc of the IEEE*, vol. 91, no. 8.

[15] H. Qi, S. S. Iyengar, and K. Chakrabarty, "Distributed sensor fusion - a review of recent research," *Journal of the Franklin Institute*, vol. 338, pp. 655–668, 2001.

[16] Y. Zhou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *IEEE Twenty-Second Annual Joint Conference of the Computer and Communications Societies*, vol. 2, no. 2.

[17] T. Wong, T. Tsuchiya, and T. Kikuno, "A self-organizing technique for sensor placement in wireless micro-sensor networks," in *18th International Conference on Advanced Information Networking and Applications*, vol. 1.

[18] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation*, vol. 1.

[19] P. Cheng, C. Chuah, and X. Liu, "Energy-aware node placement in wireless sensor networks," in *IEEE Global Telecommunications Conference*, vol. 5.

[20] N. Heo and P. K. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," in *IEEE International Conference on Wireless Communications and Networking*, vol. 3.

[21] ——, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5.

[22] ——, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. Syst., Man, Cybern. A*, vol. 35, pp. 78–92, Jan. 2005.

[23] M. L. Skolnik, *Introduction to Radar Systems*. 10: TATA McGRAW HILL.

[24] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities and challenge," in *Proceedings of The IEEE*, vol. 91, no. 8.

[25] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22.

[26] P. K. Varshney, *Distributed Detection and Data Fusion*. New York: SpringerL.

[27] J. Hu and R. Blum, "On the optimality of finite-level quantization for distributed signal detection." *IEEE Transactions on Information Theory*, vol. 47.

[28] J. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor network." *IEEE Transactions on Signal Processing*, vol. 51.

[29] B. C. R. Niu and V. P. K, "Decision fusion rules in wireless sensor networks using fading channel statistics," in *In Proceedings of CISS 2003*.

[30] S. C. A. Thomopoulos and L. Zhang, "Distributed decision fusion with networking delays and channel errors." *Information Science*, vol. 66.

[31] B. Chen and P. Willett, "Channel optimized binary quantiers for distributed sensor networks." in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*.

[32] B. Liu and B. Chen, "Joint source-channel coding for distribute sensor networks." in *In Proceedings of the 38th Annual Asilomar Conference on Signals, Systemss, and Computers, Pacific Grove*.

[33] B. C. Y. Lin and P. K. Varshney, "Decision fusion rules in multi-hop wireless sensor networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 2, pp. 475–488, Apr. 2005.

[34] T. M. I. F. Akyildiz, D. Pompili, "Underwater acoustic sensor networks: Research challenges," *Elsevier's Journal of Ad Hoc Networks*, vol. 3.

[35] Y. S. C. V. Chandrasekhar, W. K.G. Seah and H. V. Ee, "Localization in underwater sensor networks - survey and challenges," in *Proc. ACM WUWNet*.

[36] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," in *Proc. WiOpt*.

[37] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *FUSION*, 2002.

[38] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning - i," *Information Science*, vol. 8, pp. 199–249, 1975.

[39] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. Academic Press.

[40] E. Hisdal, "The if then else statement and interval-valued fuzzy sets of higher type," *Int'l. J. Man-Machine Studies*, vol. 15, pp. 385–455, 1981.

[41] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 643–658, Dec. 1999.

[42] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5.

[43] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," in *Proc. of the IEEE*, vol. 83, no. 3, Mar. 1995, pp. 345–377.

[44] Q. Liang and J. M. Mendel, "Equalization of time-varying nonlinear channels using type-2 fuzzy adaptive filters," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5.

[45] ——, "Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filters," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 12.

[46] ——, "Mpeg vbr video traffic modeling and classification using fuzzy techniques," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 1.

[47] Q. Liang, N. Karnik, and J. M. Mendel, "Connection admission control in atm network using survey-based type-2 fuzzy logic systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, no. 3.

[48] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4.

[49] L. Zhao and Q. Liang, "Energy-efficient self-organization for wireless sensor networks: A fully distributed approach," in *IEEE GLOBECOM*, Dallas, TX.

[50] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems.* Prentice-Hall.

[51] T. J. Ross, *Fuzzy Logic With Engineering Applications.* McGraw-Hill International Editions, 1997.

[52] T. S. Rappaport, *Wireless communications:principles and practice.* Prentice-Hall.

[53] G. L. Stuber, *Principles of mobile communication.* Boston : Kluwer Academic Publishers.

[54] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," in *Proceedings of the IEEE*, vol. 83, no. 3.

[55] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 83, no. 3.

[56] T. Cover and J. Thomas, *Elements of Information Theory.* Wiley.

[57] T. Berger, Z. Zhang, and H. Viswanathan, "The ceo problem [multiterminal source coding]," *IEEE Trans. Inform. Theory*, vol. 42.

[58] S. Pradhan and K. Ramchandran, "Generalized coset codes for symmetric distributed source coding," *submitted to IEEE Trans. Inform. Theory.*

[59] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "Asymmetric code design for remote multiterminal source coding," in *Proc. DCC'04.*

[60] P. W. Y. Sun and R. Lynch, "Waveform fusion in sonar signal processing." *IEEE Transactions on Aerospace and Electronic Systems.*, vol. 40, no. 2, pp. 462–477, Apr. 2004.

[61] G. X. X. Cheng, A. Thaeler and D. Chen, "Tps: A time-based positioning scheme for outdoor wireless sensor networks," in *IEEE INFOCOM 2004*, vol. 4.

[62] D. H. F. Liu, X. Cheng and D. Chen, "Range-difference based location discovery for sensor networks with short range beacons," *International Journal on Ad Hoc and Ubiquitous Computing.*

[63] M. D. A. Thaeler and X. Cheng, "itps: An improved location discovery scheme for sensor networks with long range beacons," *Journal of Parallel and Distributed Computing*, vol. 65, no. 2, pp. 98–106, Feb. 2005.

[64] J. K. J. Partan and B. Levine, "A survey of practical issues in underwater networks," in *Proceedings of the 1st ACM international workshop on Underwater networks, International Conference on Mobile Computing and Networking.*

[65] K. D. H. J.A. Catipovic, A.B. Baggeroer and D. Koelsch, "Design and performance analysis of a digital telemetry system for short range underwater channel," *IEEE Journal of Oceanic Engineering*, vol. OE-9, no. 4, pp. 242–252, Oct. 1984.

[66] G. Sandsmark and A. Solstad, "Adaptive beam forming and adaptive equalization for high-speed underwater acoustic data transmission," in *Proc. of Underwater Defence Tech. Conference*.

[67] X. Geng and A. Zielinski, "An eigenpath underwater acoustic communication channel model," in *OCEANS '95. MTS/IEEE.'Challenges of Our Changing Global Environment'*, vol. 2.

[68] W. Jobst and X. Zabalgogeazcoa, "Coherence estimates for signals propagating through acoustic channels with multiple paths," *Journal of the Acoustic Society of America*, vol. 65, no. 3, pp. 622–630, Mar. 1979.

[69] J. J.J. Caffery and G. L. Stuber, "Overview of radiolocation in cdma cellular systems," *IEEE Communications Magazine*.

[70] ——, "Subscriber location in cdma cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 406–416, May 1998.

[71] "Underwater acoustic technical guides - speed of sound in seawater," Tech. Rep. [Online]. Available: http://www.npl.co.uk/accoustics/techquides/soundseawater/speedsw.pdf

[72] "Square roots by newton's method," Tech. Rep. [Online]. Available: http://www-mitpress.mit.edu/sicp/chapter1/node9.html

[73] J. H. N. Bulusu and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine, Special Issue on Networking the Physical World*.

[74] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *WSNA'02*.

[75] W. D. Blare and Y. Bar-Shalom, "Tracking maneuvering targets with multiple sensors: Does more data always mean better estimates," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 1, pp. 450–455, Jan. 1996.

BIOGRAPHICAL STATEMENT

Haining Shu received the B.S. degree in Electronics and Information Systems from Peking University, China in 1996 and M.S. degree in Electrical Engineering from the University of Texas at Dallas in 2002. Prior to the master program, she was a System Engineer at Telecom Planning and Research Institute in Beijing, China. She is currently working toward the Ph.D degree in Electrical Engineering at the University of Texas at Arlington. Her research interests include fuzzy logic systems and applications, distributed source coding, sensor networks and collaborative radar system.